

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky a mezioborových studií

Studijní program: N2612 – Elektrotechnika a informatika
Studijní obor: 1802T007 – Informační technologie

Specifika přístupu ke studentskému portálu z mobilních zařízení

Specifics of acces to the student portal from mobile devices

Diplomová práce

Autor: **Bc. Otomar Němeček**
Vedoucí práce: doc. RNDr. Pavel Satrapa, Ph.D.

V Liberci 10. 5. 2012

Oficiální podepsané zadání – 1 dvoustrana textu

PROHLÁŠENÍ

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinností informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce.

Datum 10. 5. 2012

Podpis

PODĚKOVÁNÍ

Na tomto místě chci co nejupřímněji poděkovat panu doc. RNDr. Pavlu Satrapovi, Ph.D., za odborné vedení a nedocenitelné rady při vypracování této diplomové práce. Konzultace s ním byly velmi obohacující a přínosné.

Bc. Otomar Němeček

ABSTRAKT

Česky:

V diplomové práci se zabývám problematikou tvorby široce přístupných webových aplikací. Hlavním cílem této práce je specifikovat vlastnosti zařízení a jejich dopad na vývoj WWW stránek. Zároveň je cílem usnadnit studentům komunikaci skrze navržený studentský komunikační portál, na kterém i demonstрую jedno z možných řešení široce přístupných webových aplikací.

V první části práce specifikuji technologie pro vývoj webových stránek. V následující části se zabývám současným stavem řešení WWW aplikací a analyzuji je na bázi přístupnosti, použitelnosti a rychlosti. V kapitole další specifikuji různé druhy přístupů k multiplatformním webovým aplikacím vzhledem k nejdůležitějším vlastnostem zařízení, které mají na vývoj široce přístupných webových aplikací největší dopad. Další část pojednává o zhodnocení vlastností zařízení na bázi hardwaru, softwaru a dat. V následující kapitole uplatňuji poznatky z teoretické části práce a navrhuji architekturu systému, konstrukci studentského komunikačního portálu a strukturu databáze. V poslední části se věnuji jednotlivým segmentům portálu – jmenovitě funkční činnosti, uživatelskému rozhraní a rozdílům přístupu z různých zařízení.

Klíčová slova: WWW aplikace, vlastnosti zařízení, komunikační portál, HTML5, CSS3, Media Queries, JavaScript, jQuery, AJAX, Shibboleth, specifika, analýza, přístup, zhodnocení, návrh, programování, použití

ABSTRACT

English:

The diploma thesis deals with the issue of a widely accessible web applications. The main objective of this work is to specify the device properties and their impact on Web pages development. Another objective is to facilitate communication to students through designed student communication portal, which also demonstrates one possible approach to a widely accesible web applications.

In the first part I'm specifying technologies for website development. In the following section I'm dealing with the current state of WWW applications and I'm analyzing them based on accesibility, usability and speed. In the next chapter I'm specifying the various types of approaches to multi-platform web applications due to the most important properties of devices that have the greatest impact on development of a widely accessible web applications. The next chapter discusses the evaluation of the properties of devices based on hardware, software and data. In the following chapter I apply knowledge from the theoretical part of the thesis and propose a system architecture, student communication portal design and database structure. The last section is devoted to individual segments of the portal – namely function activities, user interface and access differences from different devices.

Keywords: WWW application, device characteristics, communication portal, HTML5, CSS3, Media Queries, JavaScript, jQuery, AJAX, Shibboleth, specifics, analysis, approach, evaluation, design, programming, use

OBSAH

PROHLÁŠENÍ.....	3
PODĚKOVÁNÍ.....	4
ABSTRAKT	5
ABSTRACT.....	6
OBSAH	7
SEZNAM OBRÁZKŮ	9
1 ÚVOD	11
2 SPECIFIKA POUŽITÝCH TECHNOLOGIÍ	13
2.1 HTML5	13
2.2 CSS3 – KASKÁDOVÉ STYL.....	14
2.3 MEDIA QUERIES.....	14
2.4 JAVASCRIPT A JQUERY 1.7.1.....	15
2.5 AJAX	16
2.6 SHIBBOLETH 2.4.3.....	17
2.6.1 Identity Provider	17
2.6.2 Service Provider.....	17
2.7 INTERNET INFORMATION SERVICES 7.5	18
2.7.1 PHP 5.4.0 - FastCGI modul pro IIS	19
2.7.2 MySQL 5.5.23 – databázový systém	20
2.7.3 Architektura a komunikace v IIS	21
3 ANALÝZA PROBLEMATIKY VIZUALIZACE	22
3.1 SOUČASNÝ STAV ŘEŠENÍ WEBOVÝCH APLIKACÍ	22
3.1.1 Vysvětlení použitých pojmů	22
3.1.2 Webové aplikace bez optimalizace	22
3.1.3 Webové aplikace optimalizované pro prohlížeče.....	23
3.1.4 Webové aplikace optimalizované pro prohlížeče a HW	24
3.1.5 Existuje univerzální řešení?	24
3.2 ANALÝZA PROBLÉMŮ OBECNĚ.....	25
3.3 ANALÝZA – PŘÍSTUPNOST.....	25
3.3.1 Nezobrazitelný webový obsah	25
3.3.2 Chybně interpretovaný webový obsah	27
3.4 ANALÝZA – POUŽITELNOST	28
3.5 ANALÝZA – RYCHLOST.....	29
3.6 KLASIFIKACE POTENCIÁLNÍCH ZDROJŮ PROBLÉMŮ	30
4 PŘÍSTUPY K DOSTUPNOSTI WEBOVÉ APLIKACE	31
4.1 OBECNÝ POPIS	31
4.2 MEDIA QUERIES – ÚPRAVA CSS STYLŮ.....	31
4.2.1 Zobrazení obsahu pomocí CSS	33
4.3 MEDIA QUERIES – ROZCESTNÍK.....	33
4.4 JAVASCRIPT – ROZCESTNÍK	34
4.5 MEDIA QUERIES – VYSOKÉ DPI.....	35
4.6 VIEWPORT.....	36
4.7 MEDIA QUERIES – VYSOKÉ DPI A DPR = 1.....	38
4.8 SROVNÁNÍ PŘÍSTUPŮ	39
5 ZHODNOCENÍ VLASTNOSTÍ ZAŘÍZENÍ	40
5.1 HARDWARE.....	40
5.1.1 Rozlišení a DPI	40
5.1.2 Operační paměť.....	40
5.1.3 Ovládání.....	41
5.2 SOFTWARE	42
5.2.1 Webový prohlížeč	42
5.2.2 Webový prohlížeč – HTML5	43
5.2.3 Webový prohlížeč – CSS3	43
5.2.4 Webový prohlížeč – AJAX	44
5.2.5 Moduly.....	44
5.3 DATA	44

5.3.1 Rychlost dat.....	44
5.3.2 Cena dat	45
6 NÁVRH WEBOVÉ APLIKACE.....	46
6.1 NÁVRH ARCHITEKTURY SYSTÉMU.....	46
6.1.1 Volba Internet Information Services	46
6.1.2 Cíl vizuálního rozvržení.....	46
6.1.3 Proč volba dvou souběžných pod-portálů?	47
6.1.4 Architektura systému	47
6.2 NÁVRH STRUKTURY PORTÁLU	48
6.2.1 Studentský komunikační portál	48
6.2.2 Cíl struktury komunikačního portálu	48
6.2.3 Struktura portálu a Shibboleth.....	49
6.3 NÁVRH STRUKTURY DATABÁZE.....	50
6.3.1 Cíl struktury databáze	50
6.3.2 Struktura databáze.....	50
7 TVORBA APLIKAČNÍCH STRÁNEK	51
7.1 PŘIHLÁŠENÍ.....	51
7.1.1 Princip přihlášení	51
7.1.2 Detekce zařízení	52
7.1.3 Rozdíly přístupu	52
7.2 UŽIVATELSKÉ ROZHRAŇÍ	53
7.2.1 Uživatelské rozhraní na mobilních zařízeních	53
7.2.2 Uživatelské rozhraní na desktopových zařízeních	54
7.2.3 Nové zprávy a události.....	54
7.2.4 Vysouvací menu.....	55
7.3 PROFIL	56
7.3.1 Princip profilů	56
7.3.2 Detekce profilu.....	56
7.3.3 Vlastní versus cizí profil	57
7.3.4 Fotografie uživatele.....	57
7.3.5 Návštěvy profilu.....	57
7.3.6 Nastavení profilu	58
7.4 PŘÁTELE	59
7.4.1 Princip přátelství na studentském komunikačním portálu.....	59
7.4.2 Skupiny	60
7.4.3 Přátelství	60
7.5 NÁSTĚNKA.....	61
7.5.1 Princip nástěnky.....	62
7.5.2 Vlastní nástěnka	62
7.5.3 Cizí nástěnka	63
7.5.4 Vkládání zpráv	63
7.5.5 Vkládání komentářů.....	63
7.6 SOUKROMÉ ZPRÁVY	64
7.6.1 Princip soukromých zpráv.....	64
7.6.2 Posílání soukromých zpráv z cizího profilu	65
7.6.3 Odpověď na soukromou zprávu	65
7.7 UDÁLOSTI	66
7.7.1 Princip událostí	66
7.8 HLEDÁNÍ.....	66
7.8.1 Princip modulu hledání	67
7.9 NEPŘÍMO SOUVISEJÍCÍ SOUČÁSTI.....	67
7.9.1 Touch ikona.....	67
7.9.2 Fonty	68
7.10 ODHLÁŠENÍ.....	68
7.10.1 Princip odhlášení.....	68
ZÁVĚR	69
SEZNAM POUŽITÉ LITERATURY	71
PŘÍLOHY NA CD	72

SEZNAM OBRÁZKŮ

OBRÁZEK 1. STRUKTURA HTML5	13
OBRÁZEK 2. KULATÉ ROHY V CSS3.....	14
OBRÁZEK 3. MEDIA QUERIES V HLAVIČCE DOKUMENTU.....	15
OBRÁZEK 4. MEDIA QUERIES UVNITŘ CSS SOUBORU.....	15
OBRÁZEK 5. IMPLEMENTACE CSS STYLU PRO LICHÉ ŘÁDKY TABULKY POMOCÍ KNIHOVNY JQUERY	16
OBRÁZEK 6. TYPICKÉ AJAXOVÉ VOLÁNÍ.....	16
OBRÁZEK 7. KOMUNIKACE V SYSTÉMU SHIBBOLETH PRO BROWSER/POST	18
OBRÁZEK 8. PODÍL WEBOVÝCH SERVERŮ NA SOUČASNÉM TRHU – NETCRAFT.COM.....	19
OBRÁZEK 9. UKÁZKA PHP SYNTAXE	19
OBRÁZEK 10. UKÁZKA MYSQL DOTAZU	20
OBRÁZEK 11. ARCHITEKTURA A PRŮBĚH KOMUNIKACE V IIS	21
OBRÁZEK 12. WEBOVÁ APLIKACE SERVIS24.CZ.....	23
OBRÁZEK 13. WEBOVÁ APLIKACE TECHNICKÉ UNIVERZITY V LIBERCI.....	23
OBRÁZEK 14. ROZDÍL ZOBRAZENÍ OPTIMALIZOVANÉ STRÁNKY SEZNAM.CZ VZHLEDEM K DPI	24
OBRÁZEK 15. TEXTOVÝ PROHLÍŽEČ LYNX – BASICCONFIG.COM.....	26
OBRÁZEK 16. ZÁSTUPNÝ OBSAH CHYBĚJÍCÍHO FLASH MODULU	26
OBRÁZEK 17. CHYBOVÉ HLÁŠENÍ INTERNET EXPLORERU PŘI NEDOSTATKU OPERAČNÍ PAMĚTI	27
OBRÁZEK 18. KATEGORIZACE ZDROJŮ MOŽNÝCH PROBLÉMŮ	30
OBRÁZEK 19. MEDIA QUERIES – ÚPRAVA CSS STYLŮ	31
OBRÁZEK 20. IMPLEMENTACE MEDIA QUERIES – ÚPRAVA CSS STYLŮ	32
OBRÁZEK 21. MEDIA QUERIES – ROZCESTNÍK	33
OBRÁZEK 22. JAVASCRIPT – ROZCESTNÍK.....	34
OBRÁZEK 23. MEDIA QUERIES – DEVICE-PIXEL-RATIO.....	35
OBRÁZEK 24. VZORCE PRO PRÁCI S OBRÁZKY NA ZAŘÍZENÍCH S VYSOKÝM DPI.....	35
OBRÁZEK 25. VIEWPORT – 160DPI	36
OBRÁZEK 26. POROVNÁNÍ WEBOVÉHO OBSAHU BEZ META TAGU VIEWPORT.....	37
OBRÁZEK 27. POROVNÁNÍ WEBOVÉHO OBSAHU S META TAGEM VIEWPORT	37
OBRÁZEK 28. VZOREC PRO VÝPOČET DEVICE-WIDTH	37
OBRÁZEK 29. MEDIA QUERIES – VYSOKÉ DPI A DPR = 1	38
OBRÁZEK 30. VZOREC PRO VÝPOČET FYZICKÝCH PARAMETRŮ DISPLEJE ZAŘÍZENÍ.....	38
OBRÁZEK 31. ROZLIŠENÍ A DPI.....	40
OBRÁZEK 32. CSS STYLIZACE ODKAZŮ.....	41
OBRÁZEK 33. POPIS OBRÁZKOVÝCH ELEMENTŮ	42
OBRÁZEK 34. DEMONSTRACE ZPŘÍSTUPNĚNÍ HTML5 POMOCÍ JAVASCRIPTU	43
OBRÁZEK 35. OPTIMALIZACE CSS3	43
OBRÁZEK 36. ARCHITEKTURA SYSTÉMU	47
OBRÁZEK 37. OBECNÁ STRUKTURA PORTÁLU A SHIBBOLETH	49
OBRÁZEK 38. STRUKTURA DATABÁZE – TABULKY A RELACE.....	50
OBRÁZEK 39. PŘIHLÁŠENÍ	51
OBRÁZEK 40. NASTAVENÍ META TAGU VIEWPORT.....	52
OBRÁZEK 41. NASTAVENÍ MEDIA QUERIES A ZOBRAZOVÁNÍ BLOKŮ	52
OBRÁZEK 42. UŽIVATELSKÉ ROZHRAŇÍ.....	53
OBRÁZEK 43. TLAČÍTKO ZPĚT.....	53
OBRÁZEK 44. UPOZORNĚNÍ NA MOBILNÍCH ZAŘÍZENÍCH – SPODNÍ LIŠTA	54
OBRÁZEK 45. UPOZORNĚNÍ NA DESKTOPOVÝCH ZAŘÍZENÍCH – HORNÍ LIŠTA.....	54
OBRÁZEK 46. UPOZORNĚNÍ NA NOVÉ UDÁLOSTI – PHP, HTML, MYSQL A JQUERY	55
OBRÁZEK 47. VYSOUVACÍ MENU – MOBILNÍ VERZE	55
OBRÁZEK 48. VYSOUVACÍ MENU – DESKTOPOVÁ VERZE.....	55
OBRÁZEK 49. PROFIL.....	56

OBRÁZEK 50. ZÁZNAM NÁVŠTĚVY PROFILU	57
OBRÁZEK 51. NASTAVENÍ PROFILU	58
OBRÁZEK 52. MYSQL DOTAZ PRO ÚPRAVU PROFILU	58
OBRÁZEK 53. PŘÁTELÉ.....	59
OBRÁZEK 54. ROLETOVÉ MENU.....	59
OBRÁZEK 55. FORMULÁŘ PRO PŘIDÁNÍ PŘÍTELE – MOBILNÍ VERZE.....	60
OBRÁZEK 56. NÁSTĚNKA	61
OBRÁZEK 57. POROVNÁNÍ FORMULÁŘE PRO VKLÁDÁNÍ ZPRÁV NA OBOU POD-PORTÁLECH	61
OBRÁZEK 58. POROVNÁNÍ ZPRÁV NA OBOU POD-PORTÁLECH	61
OBRÁZEK 59. MYSQL DOTAZ – OSOBNÍ NÁSTĚNKA	62
OBRÁZEK 60. MYSQL DOTAZ – CIZÍ NÁSTĚNKA.....	63
OBRÁZEK 61. VKLÁDÁNÍ KOMENTÁŘŮ – MOBILNÍ VERZE	64
OBRÁZEK 62. SOUKROMÉ ZPRÁVY	64
OBRÁZEK 63. FORMULÁŘ PRO ODESLÁNÍ SOUKROMÉ ZPRÁVY – MOBILNÍ VERZE	65
OBRÁZEK 64. ROZDÍL FORMULÁŘŮ PRO ODPOVĚĎ NA SOUKROMOU ZPRÁVU V OBOU POD-PORTÁLECH.....	65
OBRÁZEK 65. UDÁLOSTI.....	66
OBRÁZEK 66. HLEDÁNÍ	66
OBRÁZEK 67. MODUL HLEDÁNÍ.....	67
OBRÁZEK 68. ODHLÁŠENÍ – PHP	68

1 ÚVOD

Když přijde řeč na mobilní telefony nebo jiná mobilní zařízení, mnozí z nás by si ještě před pár lety nedokázali představit, že si na nich jednou budou prohlížet internetové stránky. Mobilní zařízení si však pomalu ale jistě již nějaký čas hledají skulinku do světa Internetu. Přes nevinné technologie typu WML jsme se nyní dostali k mnohem sofistikovanějším a plnohodnotným metodám zobrazování stránek. Pomocí HTML jsme nyní schopni zobrazit prakticky jakoukoli webovou stránku na všech moderních mobilních zařízeních s internetovým prohlížečem. Jak ale takovou stránku pro všechny možné typy zařízení uzpůsobit?

Toto téma diplomové práce jsem si vybral, protože mne fascinuje nejen grafická podoba a funkčnost internetových stránek, ale i rozvoj moderních technologií používaných ve webovém prostředí.

Chytré mobilní telefony zažívají v současnosti nečekaný rozmach. Je jen otázkou času, kdy se vývoj internetových aplikací začne soustředit převážně na tato zařízení. Mobilní telefony mají lidé stále po ruce a chtějí-li se připojit k Internetu, mají možnost prakticky kdykoli. Není výjimkou, že stránka, kterou se uživatel snaží zobrazit, není pro jeho zařízení uzpůsobená. Vývoj webu pro mobilní komunikátory tudíž není věcí zítřka, ale měli bychom se na tuto problematiku zaměřit již dnes.

V diplomové práci se mimo vlastností a specifik přístupů zařízení k webovým stránkám dále zabývám vývojem studentského komunikačního portálu (ukázková verze na <http://studenti-tul.wz.cz>) přístupného i skrze mobilní zařízení. Sociální sítě jsou fenoménem této doby. Usnadňují socializaci a komunikaci mezi uživateli. Pro studenty naší univerzity v současnosti neexistuje žádný portál, který by jim usnadnil se vzájemně poznávat, sdílet aktuální informace či si jen ukrátit volnou chvíli on-line mezi svými vrstevníky ze školy.

Tento portál jde ovšem o krok dál. Na portálu je možné členit lidi do skupin. Využití skupin je jednou z podstatných částí portálu, kdy je zaručena absolutní privátnost zpráv uživatele. Poslanou zprávu uvidí jen uživatelé ze skupiny, pro kterou je určena. Portál dále nabízí řadu klasických komunikačních nástrojů, jako jsou soukromé zprávy, veřejné zprávy, ukládání uživatelů do přátel, psaní komentářů, sledování aktivit přátel nebo návštěv atd.

To vše probíhá za oponou zabezpečeného připojení a pomocí školních uživatelských účtů, tudíž není potřeba registrace uživatelů a na místě nejsou ani obavy o bezpečí citlivých dat.

V první kapitole diplomové práce rozeberu jednotlivé technologie, které jsou esenciální pro tvorbu WWW aplikací. V druhé kapitole se budu věnovat současnému stavu řešení webových stránek. WWW stránky posléze ve třetí kapitole analyzuji a zjistím příčiny možných problémů s vizualizací. V kapitole čtvrté se budu věnovat specifikaci přístupů k multiplatformním webovým aplikacím v rámci nejdůležitějších vlastností zařízení, které vyplynou z analýzy, jež mají největší dopad na vizualizaci

webového obsahu. V páté kapitole specifikuji všechny vlastnosti zařízení, které nějakým způsobem ovlivňují tvorbu široce přístupných webových aplikací, a navrhnu, jak je optimalizovat. V šesté kapitole uplatním poznatky z teoretické části práce a demonstruji jedno z možných řešení široce přístupných webových aplikací, kdy navrhnu architekturu systému, studentský komunikační portál a strukturu databáze. V kapitole závěrečné se budu věnovat jednotlivým částem portálu, jejich funkčním činnostem, uživatelskému rozhraní a rozdílům přístupu z různých zařízení.

2 SPECIFIKA POUŽITÝCH TECHNOLOGIÍ

Vývoj internetových stránek je neprávem jednou z nejvíce podceňovaných disciplín v odvětví informačních technologií. Předsudky nezainteresované společnosti k vývoji webových aplikací jsou založené zejména na přístupnosti samotných technologií pro vývoj internetových stránek i mladším generacím. Tím vzniká všeobecná mýlka, že pro vývoj webových aplikací není potřeba nijak zvlášť odborných znalostí, ale opak je pravdou. Člověka, který žije v takové deziluzi, jistě zaskočí fakt, že pro vývoj profesionálních webových aplikací je potřeba znát mnoho technologií, které se vzájemně velmi těsně prolínají.

Záměrně nebudu čtenáře zatěžovat celou strukturou použitých technologií, ale vyberu pouze ty, které jsou pro tuto diplomovou práci stěžejní.

2.1 HTML5

HTML může být chápáno jako sada technologií či jako značkovací jazyk. V tomto případě pokud budeme hovořit o HTML5, je myšlen značkovací jazyk, který umožňuje publikaci dokumentů na internetu ve webové podobě. Vznik samotného jazyka sahá k roku 1991, kdy byla vydána verze 0.9, která ještě nepodporovala grafický režim. Zatím poslední verzi HTML je v4.01, která byla vydána v roce 1999. HTML5 je v praxi již využíváno, ovšem s jeho dokončením se počítá až během tohoto roku (2012).

Jazyk HTML je charakterizován množinou značek (tagů) a jejich atributů. Mezi jednotlivé tagy se umísťuje text a tím se určuje jejich význam. Názvy značek se uzavírají do ostrých závorek < a >. Tagy mohou být párové či nepárové. Párové značky musí být uzavřeny tagem s lomítkem.

```
<!DOCTYPE html>
<html lang="cs-cz" dir="ltr">
  <head>
    <meta charset="UTF-8">
    <title>Titulek stránky</title>
  </head>
  <body>
    <header></header>
    <nav></nav>
    <aside></aside>
    <section>
      <article></article>
    </section>
    <footer></footer>
  </body>
</html>
```

Obrázek 1. Struktura HTML5

Jedním z nejdůležitějších nových prvků HTML5 je možnost využívat více tagů, jež sémanticky lépe vysvětlují kostru webu. Konkrétně např. <header>, <aside>, <nav> či <footer>, které nejen zpřehlední kód, ale lze jich náležitě využít i při

konstruování kaskádových stylů. V současné verzi HTML4.01 je pro tyto účely užíván pouze tag <div> s příslušným identifikátorem.

Další důležitou vlastností HTML5 je rozšíření atributu `type` u tagu `input`, jež nám lépe umožňuje charakterizovat dané složky formuláře. Podrobný popis HTML najdete například v [1] a [2].

2.2 CSS3 – KASKÁDOVÉ STYLY

CSS (Cascading Style Sheets; česky Kaskádové styly) je jazyk, který popisuje vzhled jednotlivých značek dokumentu HTML, XHTML či XML. Hlavní účel CSS je oddělit vzhled dokumentu od obsahu.

CSS bývá obvykle deklarováno uvnitř tagu, v hlavičce dokumentu či v samostatném souboru, který je následně připojen k hlavičce. Jeho zobrazení však nemusí být ve všech prohlížečích stejné. Zpravidla vždy, když se vyvíjí webová aplikace, nastává problém se zobrazováním CSS v různých prohlížečích. To je dáno jak vývojem CSS, tak i jednotlivých prohlížečů a jejich chyb.

Výhody CSS jsou ovšem obrovské. Díky CSS může webový správce oddělit styly od obsahu, změnit podobu webu pomocí pár CSS stylů či ovlivňovat globálně požadované elementy. Pomocí CSS se zpřehlední kód celého webu. Důležitou vlastností CSS je jeho provázání s ECMAScriptem (JavaScript, JScript, ActionScript). Ten může ovlivňovat prvky s CSS stylem a nadále dynamicky vylepšovat vzhled webové stránky.

V dnešní době není standard CSS3 natolik rozšířen jako jeho předchozí verze. Je to dáno především z vývojového hlediska a to zejména tím, že CSS3 teprve vzniká. Mnoho prohlížečů si tak začalo vyvíjet své vlastní příkazy pro styly. Zárným příkladem mohou být např. kulaté rohy, které se musí deklarovat až čtyřmi řádky pro různé typy prohlížečů. (*Obrázek 2. Kulaté rohy v CSS3*) Styly se pro jednotlivé prohlížeče již začínají sjednocovat, ovšem stále je tu riziko, že uživatel nebude mít nejnovější verzi svého prohlížeče. Více podrobnějších informací k CSS3 v [3].

```
.kulateRohy {  
    border-radius: 10px;  
    -moz-border-radius: 10px;  
    -webkit-border-radius: 10px;  
    -khtml-border-radius: 10px;  
}
```

Obrázek 2. Kulaté rohy v CSS3

2.3 MEDIA QUERIES

Media Queries slouží k tomu, aby se na webovou stránku uplatnily různé CSS styly v závislosti na cílovém médiu a jeho vlastnostech. Díky Media Queries je možné vzhled webové stránky přizpůsobit různým zařízením.

Můžeme je definovat v hlavičce dokumentu pomocí HTML atributu s odkazem na CSS soubor anebo přímo uvnitř CSS souboru pomocí deklarace média (popř. i jeho vlastností), pro které budou dané CSS vlastnosti použity.

```
<link rel="stylesheet" type="text/css" media="screen" href="sans.css">
```

Obrázek 3. Media Queries v hlavičce dokumentu

```
@media screen {  
    * { font-family: sans-serif }  
}
```

Obrázek 4. Media Queries uvnitř CSS souboru

Media Queries jsou definované již pro předchozí verzi HTML a CSS, tudíž se není potřeba obávat o jejich funkčnost na zařízeních, která nepodporují HTML5 či CSS3. Převážně se jedná o starší zařízení, ale může se jednat například i o nadcházející technologie, které během počátečního prodeje podporu pro HTML5 nemají – např. Playstation VITA.

Mezi nejčastěji používané typy médií v deklaraci Media Queries patří `screen` a `print`. V budoucnu se pro zajímavost např. počítá i s typem `3d-glasses`.

Názvy některých vlastností médií v deklaraci Media Queries velmi nápadně připomínají pojmenování CSS vlastností. Jejich efekt je ovšem zcela odlišný. CSS vlastnosti nám určují vzhled dokumentu, přičemž vlastnosti médií v deklaraci Media Queries se ptají, zda dané zařízení splňuje námi předepsané požadavky.

Media Queries jsou v diplomové práci ještě dále podrobněji rozebírané vzhledem k řešeným problémům. Jednoduché příklady použití Media Queries nalezneme v [4]. Popis všech typů médií a jejich vlastností použitelných v deklaraci Media Queries nalezneme on-line v [5].

2.4 JAVASCRIPT A JQUERY 1.7.1

JavaScript je objektově orientovaný skriptovací jazyk, který je vykonáván na straně klienta. Je určen ke tvorbě dynamického webového obsahu. Jsou jím povětšinou ovládány dynamické prvky bez nutnosti kontaktovat server, např. vysouvací lišty, tlačítka či animace.

Významnou výhodou užití JavaScriptu je menší zátěž serveru, na kterém je webový obsah umístěn. Díky JavaScriptu lze provést úkony, kterých bychom pomocí jiných metod docílili jen velmi těžko či vůbec. Nevýhodou je otevřenost skriptů všem lidem, jelikož JavaScript je volně přístupný v HTML zdrojovém kódu stránky.

Program JavaScriptu se spouští až po samotném načtení celého webového obsahu, na rozdíl od PHP, které je interpretováno ještě před samotným stažením. JavaScript nemůže pracovat se soubory či sítí, aby neohrozil soukromí uživatele. [6]

Práci s JavaScriptem usnadňují a rozdíly jednotlivých implementací skrývají knihovny, z nichž pravděpodobně nejpopulárnější je jQuery. Klade důraz na interakci mezi HTML a Javascriptem. Její první verze spatřila světlo světa počátkem roku 2006 a byla zkonstruována Johnem Resigem.

Stejně jako CSS odděluje vzhled od struktury dokumentu, tak jQuery odděluje dynamické chování webu. Místo přímé specifikace událostí do HTML struktury stránka podporující jQuery najde požadovaný element a pozmění jeho manipulátor události. [7]

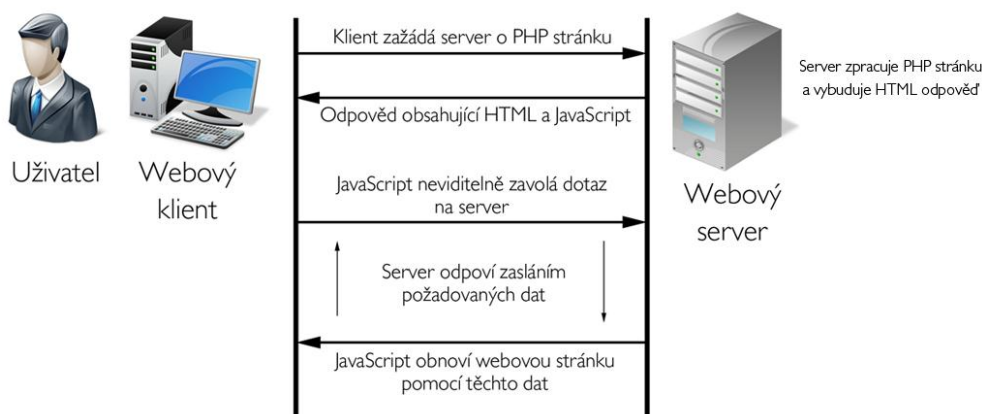
Na následujícím obrázku se pro demonstraci můžeme podívat, jak elegantně vyřešit zdánlivě složitý problém pomocí jQuery.

```
$(document).ready(function() {  
    $('tr:odd').addClass("lichyRadek");  
});
```

Obrázek 5. Implementace CSS stylu pro liché řádky tabulky pomocí knihovny jQuery

2.5 AJAX

AJAX (zkratka pro Asynchronous Javascript and XML) je název souboru technologií pro vývoj dynamických stránek zejména pomocí JavaScriptu a PHP. Užívá techniky pro dotazování serveru na pozadí webové aplikace, kdy načte nová data a obnoví tak určitá místa stránky bez nutnosti plného znovunačtení celého kódu stránky. Na následujícím obrázku se můžeme podívat na vizuální reprezentaci chování webové stránky, která podporuje AJAXové volání.



Obrázek 6. Typické AJAXové volání

AJAX poskytuje uživateli příjemnější prohlížení webových stránek. Dovoluje pracovat se stránkou mezitím co klient a server společně komunikují na pozadí. Práce s webem podporujícím AJAX tak může mnohdy připomínat běžnou desktopovou aplikaci.

Mezi výhody ovšem nepatří jen uživatelsky lepší prostředí, ale i menší zátěž serveru. Není potřeba při každém požadavku sestavovat celý HTML dokument. Zaslání dotazů pouze na požadované části stránky má tak příznivý vliv nejen na server, ale např. i na databázové systémy. Podrobnější informace o technologii AJAX v [8].

2.6 SHIBBOLETH 2.4.3

Shibboleth systém je standardizovaný open source software pro single sign-on identifikaci napříč nebo uvnitř organizačních struktur. Stačí, když uživatel prokáže svou identitu pouze jednou a následně může přistupovat ke chráněným zdrojům webové aplikace. Při další návštěvě webu se autentizace provede již automaticky.

Systém se v základu dělí na dvě části:

- ❖ Identity Provider (IdP) – software běžící na straně organizace
- ❖ Service Provider (SP) – software běžící na straně provozovatele webové aplikace

2.6.1 Identity Provider

Identity Provider (česky poskytovatel identity) je instituce, která provozuje autentizační služby. Cílem Shibboleth ideje je, aby byl uživatel autentizován u své domácí instituce i přesto, že přistupuje ke chráněným zdrojům umístěným někde jinde. IdP provozuje vlastní autentizaci uživatele a poskytuje o něm údaje. Identity provider portálu této diplomové práce je Technická univerzita v Liberci.

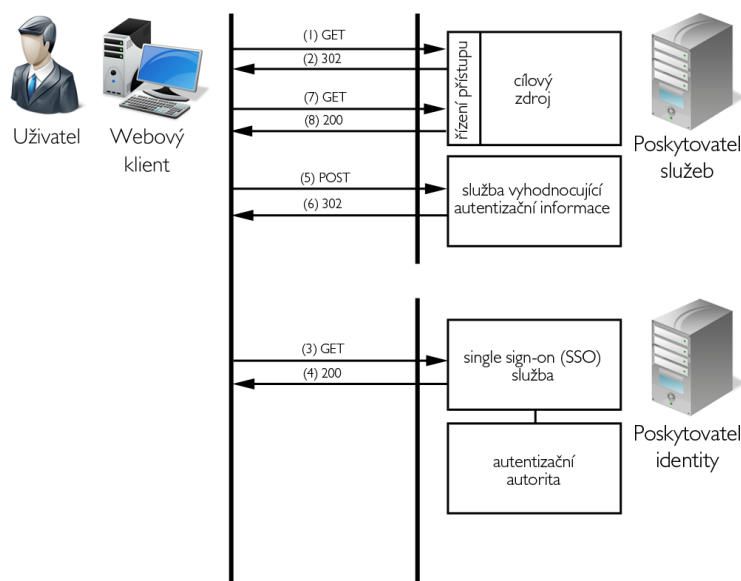
2.6.2 Service Provider

Service Provider (česky poskytovatel služeb) je pojem, který označuje samotný WWW server, o jehož stránky projevil uživatel zájem. Existují případy, kdy SP nepotřebuje o uživateli vědět žádné konkrétní informace. V takových případech je ověřena pouze identita uživatele a je mu umožněn přístup ke chráněným zdrojům webu bez vyžádání jakýchkoli dat o uživateli poskytovateli služeb. Shibboleth (IdP) tedy omezuje informace o uživateli podávaných SP. Umožňuje uživatelům i definovat údaje, které o nich smí být prozrazeny.

Další významnou výhodou Shibboleth technologie pro poskytovatele služeb je zejména umístění citlivých dat uživatele na důvěryhodném místě. Webová aplikace nemusí uchovávat veškerá data o uživateli (např. uživatelská jména a hesla) na svém interním databázovém serveru a otázku bezpečnosti přístupu k takové webové aplikaci lze přenechat na poskytovateli identity.

Roli poskytovatele služeb v této diplomové práci hraje server, na kterém je umístěn samotný studentský portál.

Technologickým základem Shibbolethu je Security Assertion Markup Language (SAML). Shibboleth tedy řeší autentizaci WWW služeb, kdy využívá profily Browser/POST a Browser/Artifact. V této diplomové práci využívám konkrétně Browser/POST. Na komunikaci v systému Shibboleth pro Browser/POST se můžeme podívat na následujícím obrázku.



Obrázek 7. Komunikace v systému Shibboleth pro Browser/POST

Počátek transakce (1) začíná požadavkem na cílový zdroj. Pokud poskytovatel služeb uživatele nezná, bude odpovědí WWW serveru přesměrování (2) na poskytovatele identity. Zpráva obsahuje informace o poskytovateli služeb, požadovaném cílovém zdroji a adresu, kam se má uživatelův klient vrátit s odpovědí.

Uživatelův klient se obrátí na poskytovatele identity (3), který ověří jeho totožnost. Jestliže se na něj klient již nedávno obracel, už tu má pravděpodobně bezpečnostní kontext, který bude využit i tentokrát - proto Single Sign-On (SSO). Výsledkem činnosti poskytovatele identity je WWW stránka s formulářem (4), který obsahuje informace pro poskytovatele služeb.

Cílová adresa pro zpracování formuláře se nachází opět u poskytovatele služeb. Zpracování formuláře zajistí služba vyhodnocující autentizační informace (5). Pokud autentizace proběhla úspěšně, vytvoří se bezpečnostní kontext a přesměruje (6) uživatele na původně požadovaný zdroj.

Při dalším požadavku na cílový zdroj (7) se zopakuje původní požadavek (1), tentokrát ale již s cookie odkazující na bezpečnostní kontext a proto bude požadovaná stránka ihned poskytnuta (8).

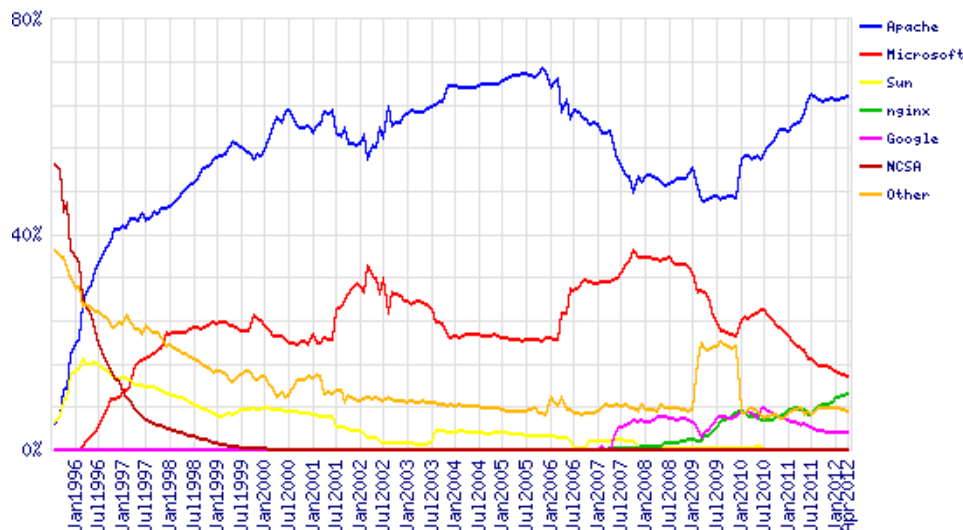
Celý tento proces se tedy vykonává pouze jednou, a to když uživatel vstupuje do určitého chráněného prostoru webové aplikace.

Čerpáno z oficiálního zdroje [9] a odborného článku [10].

2.7 INTERNET INFORMATION SERVICES 7.5

IIS je webový server vytvořený společností Microsoft pro operační systém Windows, který nabízí podporu pro rozšiřující moduly. Jedná se o nejpoužívanější webový server po serveru Apache (používá se na 13,66% všech serverů; z dubna 2012 [11]). Podporuje všechny známé webové protokoly. Ve výchozí instalaci Windows není IIS zapnut a musí se před používáním doinstalovat.

Jako každý webový server, tak i IIS umožňuje komunikaci s webovým obsahem. Jeho úlohou je zpracovat požadavky od uživatele a posílat výsledky zpět. Je schopný interpretovat jak statické, tak i dynamické stránky. Může komunikovat s PHP modulem i MySQL nebo MSSQL.



Obrázek 8. Podíl webových serverů na současném trhu – Netcraft.com

2.7.1 PHP 5.4.0 - FastCGI modul pro IIS

PHP je nejpobulárnější skriptovací programovací jazyk současnosti, který je určen zejména pro tvorbu dynamického webového obsahu. Je zpracováván na straně serveru. Skript, který programátor napíše v jazyku PHP, je běžným uživatelům utajen. Koncový uživatel je seznámen až s „hotovým“ obsahem, který prošel PHP interpretem, a který je zformulován do HTML odpovědi. V projektu je použita verze PHP 5.4.0.

Jazyk PHP je obvykle vkládán do souboru společně s HTML syntaxí (což ovšem není pravidlem), odkud je následně interpretován serverem s PHP modulem, který vygeneruje danou webovou stránku. Jako všechny Open Source programy, i PHP je jazyk nezávislý na platformě, a je volně přístupný.

Jeho výhodou je snadné prolínání kódu se značkovacími jazyky typu HTML či WML a jednoduchá komunikace s databázemi, např. MySQL, což lze využít zejména při tvorbě webových aplikací. Dále je to pak flexibilita a relativně krátká křivka osvojení.

Syntaxe jazyka spočívá v zavádění jednotlivých příkazů a proměnných uspořádaných do řádků, které musí být zakončeny středníkem. Řazení do řádků není pravidlem, nýbrž je používáno pouze pro přehlednost kódu. [8]

```
<?php
    $retezec = "Hello World!";
    print $retezec;
?>
```

Obrázek 9. Ukázka PHP syntaxe

PHP na IIS lze zavést jako ISAPI nebo (Fast)CGI modul. ISAPI je považován za rychlejší, ale méně stabilní modul a to zejména vzhledem k určitým rozšířením webového serveru jako např. některé dotazy MySQL. Z praktického hlediska se ISAPI a CGI liší ale funkcionalitou. ISAPI používá sdílený proces pro všechny návštěvníky, což vede k menší zátěži webových aplikací s mnoha návštěvami. CGI vytváří procesy pro každou návštěvu zvlášť – čím více návštěv, tím více místa paměti zabírá. Odvětví CGI je FastCGI, které místo vytváření procesů pro každou návštěvu používá jeden perzistentní proces pro správu veškerých návštěv a nemusí tak pokaždé parsovat `php.ini` a spouštět `php-cgi.exe`.

CGI modul se tedy používá všude tam, kde je zapotřebí vysoké spolehlivosti webové aplikace a ISAPI tam, kde je potřeba mít rychlejší odezvu. FastCGI přebírá spolehlivost od CGI a snaží se vyvážit rychlost implementací ideje z ISAPI. FastCGI je dokonce na webových aplikacích, které se velmi často připojují k databázi, často rychlejší než ISAPI. [12]

Pro tento projekt jsem zvolil FastCGI modul jelikož kladu důraz na stabilitu a předpokládám, že studentský portál se bude velmi často dotazovat na data z databáze.

2.7.2 MySQL 5.5.23 – databázový systém

MySQL je relační databázový systém, který spolupracuje s technologiemi PHP a IIS na zpřístupnění a zobrazení dat. Zpracovává dotazy v jazyce SQL (Structured Query Language; česky strukturovaný dotazovací jazyk), který je navržen pro práci s velkým objemem dat. V projektu používám poslední verzi MySQL 5.5.23.

Data jsou uložena v podobě tabulek, které může správce webu spojovat, dělit, hledat či měnit. Řádky jsou zpodobněny záznamy. Ve sloupcích jsou pak jednotlivé hodnoty. Výhodou MySQL je rychlost a jednoduchost. K práci s daty můžeme použít jak samostatný terminál, tak i jazyk PHP či ASP nebo další. Data získáváme pomocí dotazů, které se píšou jako jednotlivé příkazy zakončené středníkem.

```
SELECT * FROM pratele
INNER JOIN uzivatele ON pratele.pritel = uzivatele.id
WHERE pratele.uid = '1' AND uzivatele.posledni_navsteva > 1240000
ORDER BY rand() limit 0, 10;
```

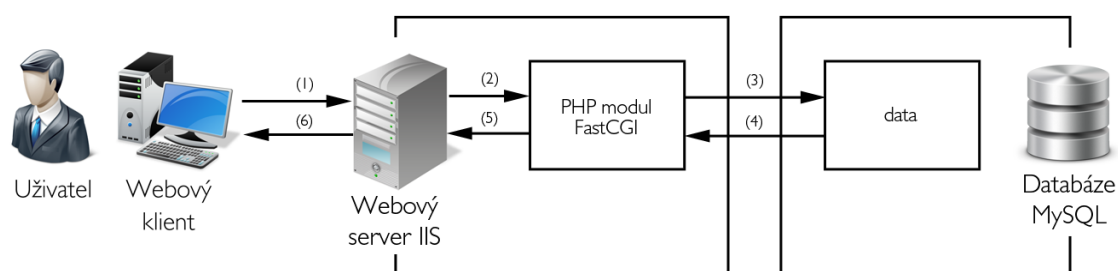
Obrázek 10. Ukázka MySQL dotazu

MySQL se nezavádí jako modul do žádného webového serveru. Jedná se o samostatnou službu, která běží souběžně s ostatními. I když se považuje za nativní databázi IIS zejména MSSQL a mnoho lidí považuje právě tuto databázi za nejvíce robustní, zvolil jsem open source MySQL, a to nejen z důvodu vlastních zkušeností, ale zejména protože předpokládám multi-platformnost studentského portálu, kdy bude moci být portál umístěn např. na serveru, kde je operačním systémem Linux. Databáze MSSQL je bezprostředně svázána s operačním systémem Microsoft Windows a její zprovoznění v OS Linux je velmi komplikované a pro praktické využívání přímo nevhodné.

Migrace dat mezi databázemi nemusí být vždy ideální. Nehledě na odlišnou strukturu dotazů, tudíž by v takovém případě musely být přepsány i některé části kódu webové aplikace.

2.7.3 Architektura a komunikace v IIS

Architektura webového serveru je založena na bázi vrstev. První vrstva je mezi webovým klientem uživatele a serverem. Následující vrstva je mezi IIS a modulem PHP. Poslední vrstva je mezi PHP modulem a MySQL databází. Na následujícím obrázku se můžeme podívat na vizuální reprezentaci takového systému.



Obrázek 11. Architektura a průběh komunikace v IIS

Komunikace se serverem začíná (1) vyžádáním webového obsahu např. vložení URI do prohlížeče nebo prostřednictvím hypertextového odkazu. Server pošle dynamický obsah stránky (2) modulu PHP. Ten společně se serverem zajistí komunikaci s MySQL (3), obdrží požadovaná data (4) a interpretuje dynamický obsah. Data se pošlou zpět IIS serveru (5) a ten vybuduje ze statických a interpretovaných dat HTML odpověď a následně ji předá (6) uživateli. Při každém dalším znovunačtení stránky nebo při přechodu na jinou stránku na témže webu se postup opakuje.

Ve skutečnosti je postup ještě složitější díky implementaci technologie Shibboleth, kdy před samotnou komunikací s webovým obsahem serveru předchází ještě autentizace uživatele. Tato vizuální reprezentace slouží zejména pro představu vnitřní komunikace webového serveru a jeho správu statického a dynamického obsahu.

3 ANALÝZA PROBLEMATIKY VIZUALIZACE

Jak jsme si již zběžně nastínili v úvodu, tak vývoj internetových aplikací je věcí značně dynamickou a variabilní. Nejsou stanovena žádná pevně daná pravidla pro vývoj webových stránek v rámci různých druhů zařízení a jejich vlastností. Nezřídka se tedy stává, že stránka, kterou chceme zobrazit, je pro naše zařízení neuzpůsobená. Tento problém většinou nastává, ač tomu není pravidlem, zejména pokud si prohlížíme WWW stránky pomocí mobilního telefonu. Uživatel může narazit na jeden nebo více problémů, které mu znepríjemní nebo úplně znemožní prohlížení webového obsahu.

V první řadě si před samotnou analýzou nastíníme současnou situaci řešení webových aplikací, její nedostatky a chyby. Následně, chceme-li předejít problémům vizualizace, musíme analyzovat, na jaké problémy může uživatel narazit a vypátrat jejich původ. Na závěr pak zdroje problémů, jež z analýzy vyplynuly, klasifikujeme vzhledem k vlastnostem zařízení a uzpůsobíme je tak pro další rozbor.

Analýza vizualizace je etapa, která by se dala označit jako základní stavební kámen pro vývoj široce přístupných webových aplikací, a která by se v žádném případě neměla opomíjet.

3.1 SOUČASNÝ STAV ŘEŠENÍ WEBOVÝCH APLIKACÍ

Webový designer začátečník, laik nebo jen designer současnosti dostatečně neznalý problematiky tvorby široce přístupných webových aplikací, si často neuvědomuje nedostatky a chyby, kterých se při vývoji WWW stránek dopouští. Často se jedná o věci, které jsou pro tvorbu široce přístupných stránek esenciální, ale jelikož neexistuje žádný soubor pravidel pro jejich tvorbu ani výčet důležitých vlastností, nelze se nad tím podívat.

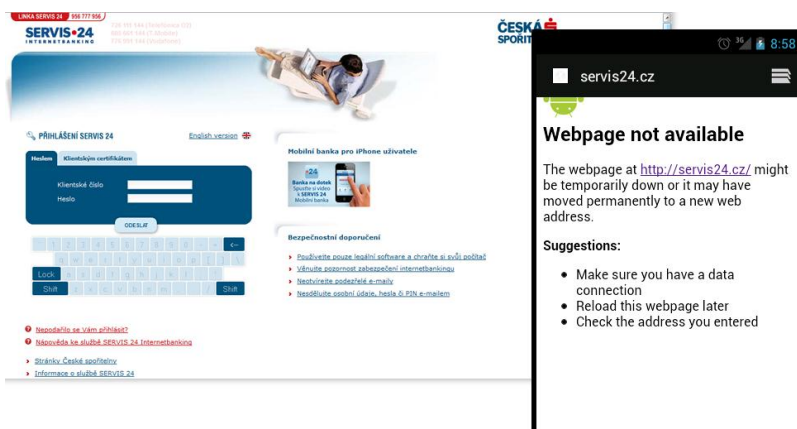
3.1.1 Vysvětlení použitých pojmů

- ❖ *Široce přístupná webová aplikace* – stránka, kterou lze bez problémů zobrazit na široké škále zařízení bez ztráty důležitého obsahu nebo čitelnosti
- ❖ *Profesionální webová aplikace* – stránka, která povětšinou slouží komerčním účelům a na její vývoj je vynaloženo značné úsilí
- ❖ *Běžná webová aplikace* – stránka, která slouží pro osobní potěšení nebo její finanční ohodnocení nebylo adekvátní k potřebnému času pro optimalizaci aplikace

3.1.2 Webové aplikace bez optimalizace

Není výjimkou, kdy se designer při vývoji webové stránky zaměří pouze na platformu, na které stránku vyvíjí. Tím je myšlen jak hardware, tak i software, který má zrovna po ruce. Autor nekontroluje vizualizaci na žádných jiných zařízeních nebo prohlížečích, kde mohou vznikat různé problémy. I když je takový přístup u profesionálních stránek již na ústupu, tak vzpomeňme společně, kdy ještě v nedávné době běželo internetové bankovníctví na stránce Servis24.cz pouze pod Microsoft Internet Explorerem (IE). Tato stránka je sice již optimalizována pro desktopové

prohlížeče, ale stále se jaksí zapomíná na existenci mobilních zařízení. Důsledek takového nedostatku je zřejmý, ztratíme uživatele nebo jej donutíme instalovat či používat hardware nebo software, který nechce a jeho zkušenost s takovou webovou aplikací v něm zajisté nezanechá dobrý dojem. Webová stránka tohoto typu nemůže být ani v nejmenším považována za široce přístupnou.



Obrázek 12. Webová aplikace Servis24.cz

3.1.3 Webové aplikace optimalizované pro prohlížeče

Dalším typem řešení stránek (již méně častým u běžných stránek, ale v současnosti obvyklým u stránek profesionálních) je, kdy designer zběžně rozumí problematice a stránku sice pořád neuzpůsobuje různorodosti hardwaru, ale už se soustředí na faktory, které s sebou přináší rozmanitost webových prohlížečů. Stránku je tak možno zobrazit na více (obvykle pár nejpoužívanějších) prohlížečích s minimálním rozdílem. Ovšem stále je uživatel svázán s hardwarovou platformou, pro kterou je webová aplikace vyvinuta. Takovýto přístup byl použit např. při vývoji WWW stránek Technické univerzity v Liberci. Ačkoli je tento typ postojů již o něco lepší vzhledem k dostupnosti webové aplikace, stále nemůžeme stránku považovat za široce přístupnou.

Na následujícím obrázku můžeme porovnat, jak taková stránka vypadá na běžném desktopovém prohlížeči v porovnání s prohlížečem na mobilním zařízení.



Obrázek 13. Webová aplikace Technické univerzity v Liberci

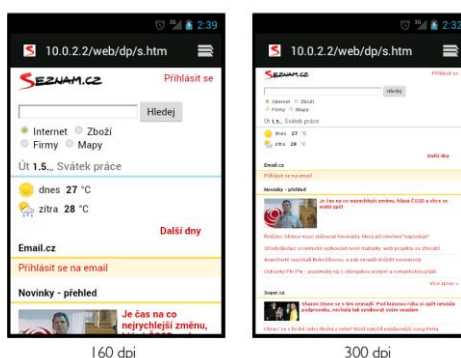
Z obrázku je patrné, že webová stránka nebyla nikterak uzpůsobována pro mobilní zařízení. Ač se nám v tomto případě zobrazila sémanticky dobře, tak je uživatel nucen stránku po načtení přibližovat, aby byl její obsah alespoň čitelný, když už není uživatelsky přívětivý.

3.1.4 Webové aplikace optimalizované pro prohlížeče a HW

Tento typ řešení u profesionálních stránek je již méně častý a u běžných webových aplikací si dovolím tvrdit, že i výjimečný. Uživateli je předložena stránka, kterou zvládne zobrazit vícero prohlížečů a její vizuální podoba je optimalizována i pro jednu nebo více dalších hardwarových platform čistě na bázi jejich rozlišení. Takovou stránku si můžeme prohlédnout jak na více prohlížečích, tak i např. na mobilním zařízení. Přístup tohoto rázu by se mohl jevit jako zcela ideální.

Existují ovšem další kritéria spjatá s vývojem stránek, na která se často zapomíná. Uzpůsobit webovou aplikaci pro různorodý hardware je totiž daleko složitější než jen připravit její layout (česky vizuální reprezentaci) pro dané zařízení, jak se tomu děje dnes. Opomíjí se aspekty jako např. DPI zařízení, RAM, ovládací prvky nebo podpora pro různé doplňkové moduly typu Flash či Silverlight.

Proto je zobrazení takových webových aplikací na různých platformách i tak často velmi rozdílné ba přímo chybné.



Obrázek 14. Rozdíl zobrazení optimalizované stránky Seznam.cz vzhledem k DPI

Z obrázku je rozdíl patrný. Máme-li zařízení se stejnými rozměry displeje, ale různým DPI, tak se optimalizovaná stránka nemusí zobrazit vždy ideálně.

3.1.5 Existuje univerzální řešení?

V současné době, kdy můžeme požadovat zobrazení webové aplikace na různých třídách zařízení, které mohou mít rozličné webové prohlížeče, nemůžeme aplikovat žádné univerzální řešení, které by bylo použitelné pro veškeré druhy stránek. Lze však na základě analýzy problémů, které se mohou vyskytnout při vizualizaci WWW aplikací, vytvořit „kuchařku“ pro tvorbu široce přístupných stránek a zmíněného efektu univerzálnosti dosáhnout.

3.2 ANALÝZA PROBLÉMŮ OBECNĚ

Při pokusu o zobrazení neoptimalizované WWW stránky může uživatel narazit na problémy, které jsem analyzoval z hlediska výskytu ve třech stupních přístupu k webové aplikaci:

- ❖ Přístupnost
- ❖ Použitelnost
- ❖ Rychlost

Výběr těchto tří pohledů jsem zakládal na osobních zkušenostech s webovými aplikacemi a na důležitosti jejich charakteru. Webová stránka by dle mé ideje měla být přístupná a použitelná na všech zařízeních bez omezení. Její rychlost vizualizace je ovšem také důležitým faktorem, jelikož uživatel je člověk pohodlný a bez patřičné rychlosti načtení aplikace by mohl o webový obsah ztratit zájem.

Přítomnost problémů spojená s nějakým z těchto faktorů je povětšinou podmíněna platformou uživatele a vzájemně se nemusí vylučovat.

Bližší informace o těchto třech přístupech k webové stránce může čtenář nalézt v informačních zdrojích [13], [14] a [15]

3.3 ANALÝZA – PŘÍSTUPNOST

Na přístupnost webové aplikace se můžeme v dnešní době dívat ze dvou úhlů. Přístupnost WWW aplikace z různých typů zařízení nebo uzpůsobení přístupnosti WWW aplikace pro lidi s nějakým handicapem. Pokud budu v diplomové práci hovořit o přístupnosti, mám tím na mysli přístup k aplikaci z různých zařízení s různým softwarem.

V současnosti vyvstává v rámci přístupnosti z různých zařízení mnoho problémů, které můžeme v základu rozdělit do dvou kategorií:

- ❖ Nezobrazitelný webový obsah
- ❖ Chybně interpretovaný webový obsah

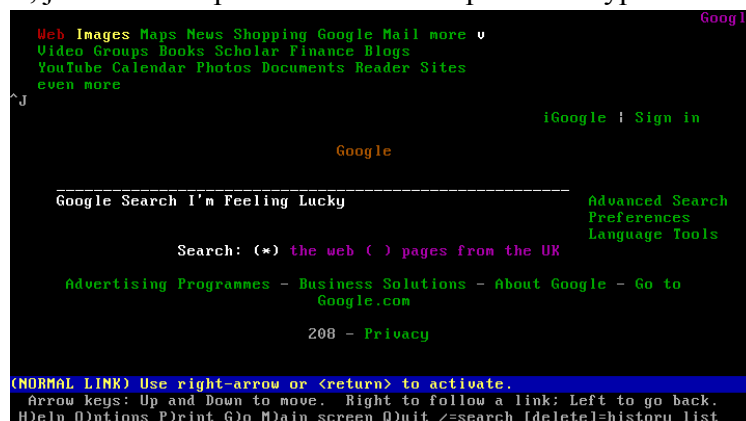
3.3.1 Nezobrazitelný webový obsah

Tato kategorie zahrnuje několik činitelů, které mohou zásadně ovlivnit chod webové aplikace:

- ❖ Typ prohlížeče
- ❖ Moduly a JavaScript
- ❖ Operační paměť zařízení

Jakmile při tvorbě WWW stránky špatně odhadneme možnosti zařízení a jeho vztah k těmto činitelům, může se nám stát, že se části stránky uživateli nezobrazí a stránka ztratí důležité informační části nebo v horším případě se stránka nezobrazí vůbec.

Problém, který může nastat s použitím typu prohlížeče (jež v základním pojetí dělíme na textové a grafické) může nastat se zobrazováním obrázků, kdy jeden uživatel obrázky vidět může a druhý nikoliv. Jedná se o jeden z typů problémů, které jsou částečně řešitelné na aplikační bázi programátorem. Na následujícím obrázku se můžeme podívat, jak webová aplikace v textovém prostředí vypadá.



Obrázek 15. Textový prohlížeč Lynx – Basicconfig.com

Extrémní případ problému s typem prohlížeče může nastat např., pokud stránka obsahuje pouze jediný velký obrázek a na něm veškerý informační obsah. Textový prohlížeč by tak stránku nezobrazil vůbec.

Moduly, někdy též nazývané doplňky nebo pluginy, jsou software instalovaný na přání uživatele. Ve své podstatě rozšiřují funkčnost webového prohlížeče. Mezi známé moduly patří např. Adobe Flash, Microsoft Silverlight (nebo jeho verze pro Linux – Moonlight) či Java Applety.

Problém s těmito moduly je zřetelný již v samotné definici. Někteří uživatelé tento software nemají. Důvody mohou být dva, uživatel daný modul na svém zařízení mít nechce nebo jeho zařízení či webový prohlížeč daný modul nepodporuje.

Spoléhat se na takové moduly při vývoji webové aplikace přístupné pro širokou škálu zařízení je víceméně nevhodné.

Ve stejné spojitosti se lze zmínit i o technologii JavaScript, kterou uživatel může dobrovolně vypnout. Její přítomnost je ovšem jednou z esenciálních potřeb pro tvorbu široce přístupných aplikací, proto se na ní musíme v určitých situacích spoléhat.



Obrázek 16. Zástupný obsah chybějícího Flash modulu

Operační paměť (RAM) je neméně důležitou částí zařízení, na kterou je potřeba brát ohled. I když by se mohlo zdát, že v současnosti mají zařízení operační paměť dostatečnou pro zobrazení jakéhokoli webového obsahu, není tomu tak vždy.

Např. konzole Playstation Portable (PSP) pro internetový prohlížeč vyhrazuje místo v operační paměti mezi 4 až 32MB. Vysoce multimediální či jinak komplexní aplikaci ve webové podobě bychom pro takové zařízení tvořili jen velmi obtížně.

Problém tedy nastává v okamžiku, kdy datová kapacita webového obsahu převyšuje kapacitu operační paměti vyhrazené zařízením. Zpravidla každé zařízení v takovou chvíli vyhodí chybu „out of memory“ (česky nedostatek paměti) a webový obsah, který je nad tímto limitem zůstane nezobrazen.



Obrázek 17. Chybové hlášení Internet Exploreru při nedostatku operační paměti

Takový problém ovšem nehrozí jen na mobilních zařízeních, kde je potřeba dbát zejména na multimediální obsah, ale může nastat i na PC např. díky únikům paměti příčinou špatně napsaného skriptu v JavaScriptu, který běží na straně webového klienta.

3.3.2 Chybně interpretovaný webový obsah

Druhá kategorie, která obsahuje činitele způsobující problémy s vizualizací, se skládá z těchto částí:

- ❖ Typ prohlížeče
- ❖ Rozlišení displeje
- ❖ DPI

Tyto faktory jsou nejdynamičtější částí vizualizace WWW stránek. Špatným přístupem k vlastnostem těchto činitelů se může stát, že aplikační stránka bude uživateli zobrazena jinak, než bylo původním záměrem.

Ač byl typ prohlížeče zmíněn již v kategorii první, lze se na tento faktor zaměřit i z jiného hlediska. Můžeme je dělit i dle podpory technologií pro vývoj webových aplikací.

Problém může nastat, např. pokud píšeme aplikaci v HTML5 a neuzpůsobíme ji pro starší prohlížeče, které tuto technologii nepodporují. Prohlížeč tak nedokáže rozeznat elementy, které nezná a chybně nám vizualizuje jejich obsah.

Další, již více známý problém, se vyskytuje při práci s CSS. Každý prohlížeč nám různě vyhodnocuje vizualizaci daného stylu. Tím vznikají určité odlišnosti při vizualizaci elementů na různých prohlížečích.

V neposlední řadě nastává problém s technologií AJAX, kdy je v prohlížečích na mobilních zařízeních vyžadováno zpřístupnění JavaScriptové části přímo v souboru zobrazované WWW stránky a nelze ji oddělit do souboru samostatného. Hrozí tak chybné nebo žádné zobrazení dynamických dat.

Rozlišení zařízení hraje další zásadní roli při zobrazování stránky. Není-li webová aplikace uzpůsobena pro rozlišení např. mobilních zařízení, je možnost, že se zobrazí zcela jinak, než bylo prvotním úmyslem webového designera.

DPI daného zařízení (dots per inch; česky obrazové body na palec) jde ruku v ruce s jeho rozlišením. I když designer správně navrhne aplikaci pro různé typy rozlišení zařízení, může nastat problém, kdy bude obsah WWW stránky málo čitelný či špatně zobrazený a to např. díky vysokému nebo nízkému DPI zařízení.

3.4 ANALÝZA – POUŽITELNOST

Nebo také přehlednost a intuitivnost je pohled na webovou aplikaci z hlediska uživatele. Zahrnuje možnosti ovládání v rámci použitého zařízení, ovládací prvky webové aplikace a zobrazení vizuálního rozvržení vzhledem k intuitivnosti a snadnému zaučení uživatele.

Bez analýzy z hlediska použitelnosti by mohla být uživateli předsunuta WWW stránka, kterou by nemohl nebo neuměl ovládat.

Na použitelnost v rámci zařízení se lze dívat ze dvou úhlů:

- ❖ Zařízení bez dotykového ovládání
- ❖ Zařízení s dotykovým ovládáním

První kategorie jsou zařízení, která se obvykle ovládají pomocí směrových šipek nebo myši. Ovládání takové aplikace tkví v tom, že uživatel najede nad element, se kterým chce komunikovat anebo jej stisknutím aktivuje a provede se požadovaná akce.

Druhá kategorie obsahuje zařízení, která se ovládají jen pomocí dotyku stylusem nebo prstem. Uživatel v takovém případě nemůže komunikovat s webovou aplikací najetím nad element, ale je omezen pouze na stisknutí.

V této době, kdy je většina mobilních zařízení dotykových, je důležité při vývoji webových aplikací dbát na obě možnosti ovládání a použitelnost stránky náležitě uzpůsobit.

Uživatelské rozhraní desktopových a mobilních aplikací je velmi odlišné. Na desktopových zařízeních většinou inklinuje ke složitým dialogům s řadou ovládacích prvků, zatímco na zařízeních mobilních je výrazně jednodušší, často rozděleno na části nebo fáze.

Použitelnost vzhledem k vizuálnímu rozvržení je individuální pro každý typ webové aplikace a zařízení, na kterém je vyobrazena. V zásadě lze ale zmínit pár základních pravidel, kterých je dobré se při vývoji držet:

- ❖ Udržení jednoduchosti aplikace
- ❖ Optimalizace ovládacích prvků
- ❖ Umožnění snadné orientace v aplikaci
- ❖ Snadné a vnímavé hledání obsahu
- ❖ Intuitivní interakce s aplikací
- ❖ Využití potenciálu ovládání

3.5 ANALÝZA – RYCHLOST

Rychlost je poslední analytický pohled na webovou aplikaci. Rozumí se jí rychlost zobrazení WWW stránky na webovém prohlížeči uživatele. Problémy, jež mohou být spojovány právě s rychlostí stránky, můžeme hledat ve třech faktorech:

- ❖ Velikost dat
- ❖ Uložení dat
- ❖ Výpočetní náročnost

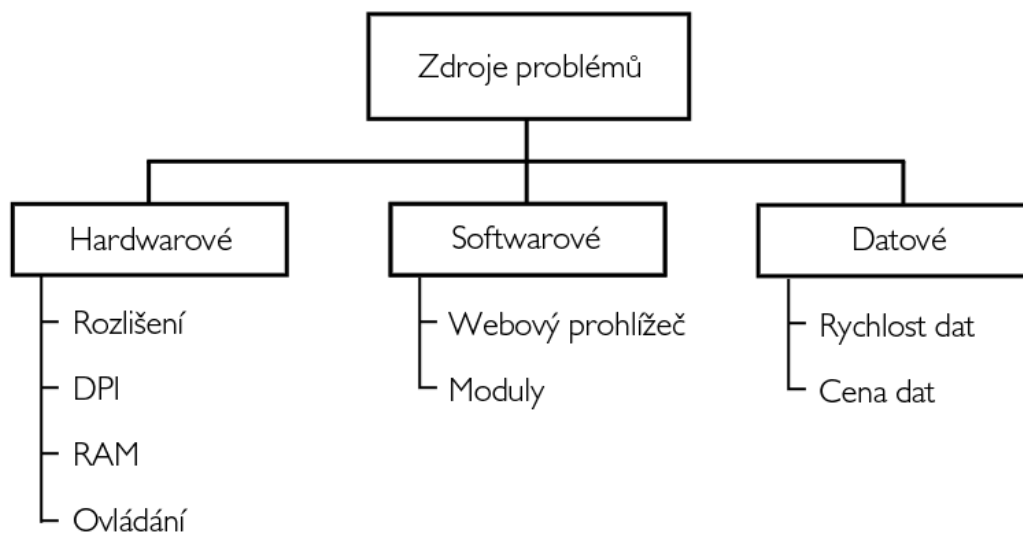
Velikostí dat je rozuměno množství přenesených dat od serveru k uživateli. Jedná se tedy o velikost HTML odpovědi serveru, jež musí webový prohlížeč zpracovat. Zpoždění obvykle nevzniká samotným zpracováním dat (málokterá webová aplikace bývá tak výpočetně náročná, aby způsobila problém tohoto rázu), ale je povětšinou spojováno s přenosem dat. Přenosové rychlosti jsou zejména na mobilních zařízeních pořád velmi nízké.

S velikostí dat je úzce spojen faktor ceny dat, i když nemá co dočinění se samotnou rychlostí, tak nám také ovlivňuje vývoj webové aplikace. U roamingu stále platí, že čím více dat přenášíme, tím dražší cenu musí uživatel svému poskytovateli připojení za přenesená data zaplatit. Má-li uživatel nějaké paušální připojení k internetu, tak je zase vázán na datový „strop“, který nemá v plánu překročit. Faktor ceny dat se týká zejména uživatelů, kteří zobrazují internetové stránky na mobilním zařízení, kde jsou ceny za přenesená data stále hodně dramatické.

Uložení dat je myšlen jak výběr databázového serveru, tak i místo fyzického uložení. V tomto případě se jedná o dobu, která je nutná ke vložení nebo vyjmutí dat z databáze a předání jejich hodnot PHP interpreteru. Problém rychlosti tedy nenastává na zařízení, které zobrazuje danou webovou stránku, ale týká se serveru, kde je umístěna databáze webové aplikace, a který se následně stará o zpracování její HTML odpovědi.

3.6 KLASIFIKACE POTENCIÁLNÍCH ZDROJŮ PROBLÉMŮ

Po analýze možných problémů musíme nyní rozdělit jejich zdroje do kategorií, se kterými lze dále pracovat a uzpůsobovat je pro WWW aplikaci. Zdroje jsem klasifikoval na bázi jejich charakteru na hardwarové, softwarové a datové. Jejich kategorizaci nejlépe vystihne následující grafické znázornění.



Obrázek 18. Kategorizace zdrojů možných problémů

4 PŘÍSTUPY K DOSTUPNOSTI WEBOVÉ APLIKACE

Ke zdárnému dosažení stanoveného cíle, kterým je uzpůsobit webovou aplikaci k zobrazení na široké škále zařízení, musíme vyřešit příčiny problémů, které při vizualizaci mohou nastat. Ovšem před samotnou optimalizací jsem specifikoval druhy přístupů k multiplatformním webovým aplikacím, které se vztahují k nejdůležitějším částem zařízení – rozlišení a DPI. Práce s těmito veličinami je velmi variabilní a jejich uzpůsobování není na první pohled zřejmé, jako je tomu u ostatních faktorů.

4.1 OBECNÝ POPIS

Jako hlavní cíl jsem si stanovil vymyslet takový přístup k multiplatformním webovým aplikacím, kterým bych bezpečně a spolehlivě dokázal specifikovat displej zařízení, abych poté mohl různým zařízením zobrazit různě uzpůsobený webový obsah.

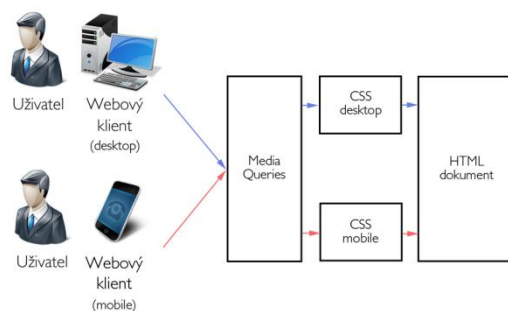
Dalším důvodem pro specifikaci druhů přístupů je stále se rozšiřující trend zmenšování rozměrů pixelů na (zejména) mobilních zařízeních. Webový obsah se může na zařízeních s vysokým DPI zobrazit buď nečitelně, nebo webová stránka nemusí dostatečně využívat potenciál takového displeje.

Prostředky pro specifikaci přístupů mi byly notebook HP Pavilion dv6, mobilní emulátor v oficiálním Android SDK a mobilní telefon Samsung Galaxy Ace.

Během specifikace jsem vždy postupoval tak, že jsem zprvu vymyslel experimentální přístup k aplikaci a pak jsem testoval jeho „neprůstřelnost“ vzhledem k různým typům rozlišení displeje, rozlišení prohlížeče a DPI zařízení. Dále jsem do výsledků zahrnul i možné problémy s datovými nároky takového přístupu v rámci různých zařízení a složitost tvorby takového přístupu.

4.2 MEDIA QUERIES – ÚPRAVA CSS STYLŮ

První experimentální přístup, který jsem provedl vzhledem k rozlišení displeje a zařízení, se konal na bázi Media Queries. Idea tohoto přístupu vychází z definice oddělení struktury webové aplikace od stylů. Obsah WWW stránky zůstává pro veškerá zařízení stejný a je pouze jinak vizualizován pomocí CSS stylů. Webový prohlížeč dle rozměrů svých nebo daného zařízení aplikuje požadované styly pro elementy HTML dokumentu.



Obrázek 19. Media Queries – Úprava CSS stylů

Tento způsob řešení přístupu k webové aplikaci je vhodný, není-li stránka datově náročná, a kdy je její obsah rovnoměrně důležitý pro všechna zařízení. Obvykle se jedná o jednoúčelové stránky sloužící např. pro prezentaci nového produktu nebo blogy a portfolia. Nelze ale vyloučit použití této metody i u komplexnějších stránek, kde může být využití Media Queries také vhodné.

Strukturu tohoto experimentálního přístupu vyjadřuje *Obrázek 19. Media Queries – Úprava CSS stylů*.

Způsob takové implementace můžeme provést skrze následující Media Queries dotaz.

```
@media screen and (max-width: 600px),
    screen and (max-device-width: 480px) {
    /* CSS pro mobilní zařízení */
    ...
}
@media screen and (min-width: 601px) {
    /* CSS pro desktop */
    ...
}
/* Defaultní CSS pro všechna zařízení */
...
```

Obrázek 20. Implementace Media Queries – Úprava CSS stylů

V kódu výše jsem pro jednoduchost použil pouze vlastnosti `width` a `device-width`, které se dotazují na šířku WWW prohlížeče a zařízení v pixelech. Pracovat ovšem můžeme i s `height` (`device-height`), jež slouží pro zjišťování výšky nebo dále rozšířit dotaz pro mobilní zařízení, zda je držíme na výšku či šířku pomocí `orientation`.

Výsledek tohoto experimentálního přístupu k aplikaci:

- ✓ Rychlá implementace oproti ostatním experimentálním přístupům
- × Jednotný webový obsah pro všechna zařízení
- × Vzhled dokumentu určujeme jen pomocí rozlišení zařízení a prohlížeče
- × Nebereme v potaz DPI

Jednotný webový obsah může být občas výhodou, ale ve většině případů na mobilním zařízení nepotřebujeme zobrazit úplně identický obsah jako na zařízení desktopovém. Značná výhoda tohoto experimentálního přístupu tkví v jeho poměrně snadné a rychlé implementaci. Nevýhodou je pak to, že jednotlivá zařízení specifikujeme pouze na bázi jejich rozlišení a nebereme v potaz jejich DPI, tudíž se tak může stát, že špatně rozeznáme druh zařízení.

Jmenovitě v tomto případě stačí, když bude mít mobilní zařízení displej se šířkou větší než 600 pixelů a tento experimentální přístup k aplikaci nebude správně fungovat.

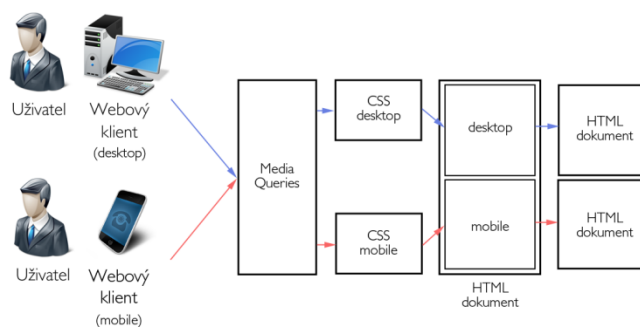
4.2.1 Zobrazení obsahu pomocí CSS

Vlastností CSS `display` můžeme zakázat zobrazení určité části webového obsahu. Vizualizovaný obsah tedy již nezůstává stejný, ale pomocí CSS vybereme pouze tu část HTML struktury, kterou pro dané zařízení chceme zobrazit.

4.3 MEDIA QUERIES – ROZCESTNÍK

Nevýhody, které plynou z předchozího přístupu, jsou viditelné na první pohled. Webový prohlížeč musí stále načítat všechny HTML elementy a jejich obsah i přesto, že jejich část může být pro vizualizaci nepotřebná.

Můžeme však využít jeho potenciál a přenést jej o úroveň výš. Aplikací předešlého postupu můžeme vytvořit tzv. rozcestník, kdy bude uživatel poté přesunut, ať už automaticky nebo pomocí přihlašovacího formuláře, na další část webu, která je již přizpůsobená pouze pro jeho zařízení a načítání zbytečných dat se tak stane minulostí.



Obrázek 21. Media Queries – Rozcestník

První HTML dokument (zleva) na obrázku hraje roli rozcestníku, kdy se jedná o jednoduchou vstupní stránku, která slouží k volbě zobrazovací varianty (a případně k přihlášení). Uživatel je tak poté přesunut na HTML dokument, který je uzpůsoben pro jeho zařízení. Po celou dobu strávenou na webové aplikaci v této části již přetrvává.

Idea tohoto způsobu řešení vizualizace (společně s ošetřením DPI, které dále zmíním) byla využita v praktické části této diplomové práce při tvorbě WWW aplikace. Tento způsob je upotřeben zejména kvůli rozdílnosti požadavků uživatelů přistupujících ke studentskému portálu z různých zařízení vzhledem k ovládání aplikace a také k udržení její jednoduchosti.

Výsledek tohoto experimentálního přístupu k aplikaci:

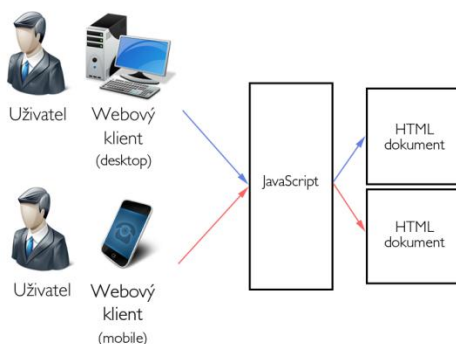
- ✓ Různý obsah pro různá zařízení
- ✓ Středně rychlá implementace
- × Vzhled dokumentu určujeme jen pomocí rozlišení zařízení a prohlížeče
- × Nebereme v potaz DPI

V tomto experimentálním přístupu k webové aplikaci jsem zajistil to, že již není potřeba, aby zařízení načítala webový obsah všech platforem. Ani tento způsob řešení přístupu není velmi časově náročný, ale tvorba dvou „oddělených“ webových stránek s sebou již přináší komplikace s tvorbou. Vývoj se tak může časově velmi protáhnout, záleží na složitosti dané aplikace. Tento přístup také ve svém základu nezajišťuje rozlišení platforem na bázi DPI, nevýhody s tím spojené plynou z předchozích experimentů.

4.4 JAVASCRIPT – ROZCESTNÍK

Rozeznávání zařízení pomocí JavaScriptu je již historický přežitek, připadá mi ovšem správné přístup k webové aplikaci s touto technologií alespoň nastínit. Principiálně funguje na bázi rozeznávání šířky prohlížeče, ale jeho datová velikost a složitost nabývá mnohem větších rozměrů, než je tomu u Media Queries. JavaScript tedy může přistupovat k DOM objektům dokumentu, v nich se ale nenachází hodnoty vlastností zařízení, takže optimalizovat nemůžeme např. pro DPI nebo rozlišení displeje (je to sice možné použitím různých triků s ne vždy optimálními výsledky, ale odložme magii stranou). Dalším případem, kdy není vhodné tento postup použít, může být, chceme-li optimalizovat webovou aplikaci i pro webové prohlížeče, jež v základu JavaScript nepodporují.

Funkce takového skriptu tkví v okamžitém přesměrování WWW prohlížeče na jinou stránku.



Obrázek 22. JavaScript – Rozcestník

Po přesměrování na požadovanou WWW stránku pak můžeme její obsah náležitě optimalizovat pro daná zařízení. Uživatel i v tomto případě řešení rozcestníku též zůstává po celou dobu setrvání na webové aplikaci již pouze v části určené pro jeho zařízení.

Výsledek tohoto přístupu k aplikaci:

- ✓ Různý obsah pro různá zařízení
- × Vzhled dokumentu určujeme jen pomocí rozlišení zařízení a prohlížeče
- × Nebereme v potaz DPI
- × Nefunguje na platformách bez aktivního JavaScriptu
- × Složitější implementace

I když tento přístup k webové aplikaci splňuje požadavek zobrazení různého webového obsahu různým zařízením, tak stále nedbá na DPI zařízení. Jestliže uživatel vypne ve svém prohlížeči zpracování JavaScriptu, tak tento přístup k webové aplikaci dokonce nebude fungovat. Celkově je implementace tohoto přístupu daleko složitější, než tomu bylo v předchozích případech a není vhodná.

4.5 MEDIA QUERIES – VYSOKÉ DPI

Předchozí experimentální přístupy jsem věnoval rozlišení zařízení bez ohledu na jejich DPI. To může vést k nedostatečnému využití potenciálu displeje na zařízeních s vysokým DPI. Pokud se nám do ruky dostane zařízení, jež má větší DPI než je v současnosti u mobilních zařízení zvyklostí (160dpi), je možné, že výrobce na takovém zařízení již vizualizaci webového obsahu uzpůsobí jeho automatickým zvětšením.

Vlastnost `device-pixel-ratio` (DPR) nám udává počet pixelů, které jsou použity pro zobrazení jednoho CSS pixelu. Např. máme-li zařízení, které zvětšuje webový obsah 2x (iPhone 4 Retina display s 326dpi), tak se pro zobrazení jednoho CSS pixelu na displeji použije 2x2 pixelů, přičemž se na iPhonu 4 dosáhne efektu 163dpi.

```
@media screen and (min-device-pixel-ratio: 1.5) {  
  /* CSS pro zařízení s vysokým DPI - Android/Opera Mobile */  
  ...  
}  
  
@media screen and (min-device-pixel-ratio: 2) {  
  /* CSS pro zařízení s vysokým DPI - iPhone4 */  
  ...  
}
```

Obrázek 23. Media Queries – device-pixel-ratio

I když se ostrost textu v takovém webovém prohlížeči uzpůsobuje automaticky, tak stále přetrvává problém se zobrazováním obrázků.

Chceme-li zobrazit obrázek na takovém zařízení 1:1 pixelu, musíme pořídit dvakrát větší, než jaký ve skutečnosti máme v plánu prezentovat a zmenšit jeho velikost v CSS o 50%. Stejná matematika platí i u zařízení, jež mají DPR 1,5 (Android Super AMOLED). V tomto případě ovšem musíme pořídit obrázek větší 1,5 krát a zmenšit jej o 66,67% abychom dosáhli efektu 1:1 pixelu. Vyplyvají nám tak následující vzorce:

$$NR = PR \times DPR$$
$$CSSz = \frac{1}{DPR}$$

Obrázek 24. Vzorce pro práci s obrázky na zařízeních s vysokým DPI

NR – nový rozměr DPR – device-pixel-ratio

PR – původní rozměr CSSz – CSS zmenšení

Ve skutečnosti se nám může stát (ač je to výjimkou), že DPR může nabývat jakýchkoli hodnot v intervalu $\langle 1; 2 \rangle$. Zařízení s nějakým takovým kuriózním DPR pak

odkazujeme dle Media Queries na obrázek, který je jeho DPR nejbližší. Tím docílíme stále vyšší ostroty, než kdybychom jej neupravovali vůbec.

Aplikujeme-li tento přístup k DPI na *4.3 Media Queries – Rozcestník*, vyplynou nám následující výsledky:

- ✓ Různý obsah pro různá zařízení
- ✓ Správné zobrazení obrázků na zařízeních s vysokým DPI
- × Vzhled dokumentu určujeme jen pomocí rozlišení zařízení a prohlížeče
- × Složitá implementace
- × Správná specifikace zařízení funguje pouze na moderních platformách

Implementací tohoto postupu docílíme na zařízeních s vysokým DPI správného zobrazení obrázků a využijeme tak potenciál displeje zařízení s vysokým DPI. V tomto případě ovšem počítáme s tím, že zařízení s neobvyklým nebo velkým DPI automaticky přiblíží webový obsah.

4.6 VIEWPORT

V předchozím příkladu jsem se zabýval ideálním případem DPI daného zařízení, kdy je obsah „zvětšen“ automaticky. Může ovšem nastat situace, kdy zařízení neupravuje obsah na 160dpi a mohou se tak vynořit komplikace se zobrazením. V takovém případě nám poslouží metadata `viewport` v hlavičce HTML dokumentu.

```
<meta name="viewport"
      content="width=device-width; initial-scale=1.0;"
/>
```

Obrázek 25. Viewport – 160dpi

Nastavením šířky obsahu na `device-width` docílíme u mobilních zařízení efektu 160dpi (obsah se automaticky upraví na šířku 320 CSS pixelů). Nesmíme zapomenout ani na `initial-scale=1.0`, které nám zakáže „prokládání“ pixelů způsobené DPR a text tak bude vyhlazený. Díky tomuto meta tagu jsme schopni stále rozdělovat zařízení pomocí původních Media Queries a rozměrů jejich šířky, protože se nemůže stát, že by se nám objevilo mobilní zařízení s větší šířkou než 320 pixelů.



Obrázek 26. Porovnání webového obsahu bez meta tagu viewport



Obrázek 27. Porovnání webového obsahu s meta tagem viewport

Vzorec pro výpočet device-width je následující:

$$dw = \frac{widthP}{DPR}$$

Obrázek 28. Vzorec pro výpočet device-width

dw – device-width widthP – počet reálných pixelů DPR – device-pixel-ratio

I když použijeme meta tag `viewport`, tak nám stále pro práci se zařízeními, které mají vysoké DPI, zůstává Media Query vlastnost `device-pixel-ratio`, dle které můžeme ostrost obrázků dále doladit.

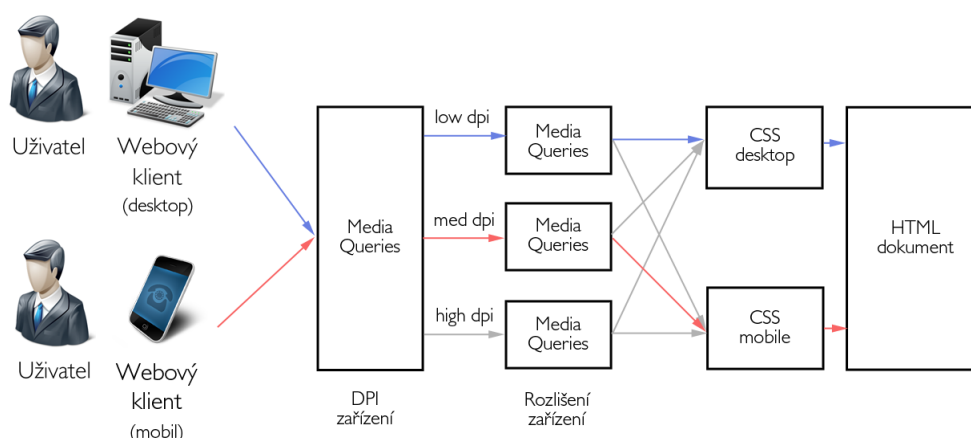
Výsledek tohoto přístupu je prakticky identický jako v předchozím případě.

Problematika DPI ovšem může být ještě složitější vzhledem k tomu, že existují „nemobilní“ nebo starší zařízení, které tento postup nemůže pokrýt. Taková zařízení nám jako `device-width` vracejí jejich skutečný počet pixelů, jelikož jejich `DPR = 1`.

4.7 MEDIA QUERIES – VYSOKÉ DPI A DPR = 1

Poslední z řady experimentálních přístupů k webové aplikaci se váže k takovým zařízením, která pixely neprokládají a mají i přesto vysoké DPI. Základ tohoto přístupu je opět na bázi Media Queries, kdy pouze rozšiřujeme již zmíněný proces kategorizace zařízení dle rozlišení.

Před samotnou klasifikací zařízení dle jejich rozlišení můžeme použít vlastnost Media Queries `resolution`, která nám porovnává DPI. Toho lze využít pro další dělení dotazu, kdy už se pak ptáme jaké rozlišení má zařízení s tímto DPI. Poté již aplikujeme požadovaný CSS styl a stránku tak můžeme náležitě uzpůsobit.



Obrázek 29. Media Queries – vysoké DPI a DPR = 1

Komplexnost toho způsobu řešení může nabývat v praxi monstrózních rozměrů, pokud specifikujeme layout (vizuální rozvržení) stránky v CSS pixelech, protože bychom pro každé zařízení s různým DPI a jeho rozlišením museli specifikovat vlastní CSS styly.

Graf výše počítá se stylizací HTML elementů a fontů pomocí relativních veličin – em nebo procent. Stránka navržená pomocí relativních veličin se prakticky zobrazí identicky na všech zařízeních s displejem stejných fyzických rozměrů i přesto, že mají různé DPI.

Pomocí DPI a rozlišení displeje můžeme vypočítat jeho fyzické rozměry dle následujícího vzorce:

$$cm = \frac{width}{DPI} \times 2,54$$

Obrázek 30. Vzorec pro výpočet fyzických parametrů displeje zařízení

cm – velikost v centimetrech
width – šířka displeje v pixelech

DPI – DPI zařízení (počet bodů na palec)
2,54 – konstanta (1 palec = 2,54cm)

V praxi tento vzorec nemusíme během rozlišování zařízení používat. Stačí předem vypočítat hodnoty šířky displeje v pixelech dle požadované fyzické velikosti a rozmyslet se, zda takové zařízení zařadíme do mobilních, desktopových či jiných

zařízení a přidělit tak jeho WWW prohlížeči požadovaný CSS soubor pouze na bázi jeho DPI a šířky v pixelech.

Dovolím si tvrdit, že webová aplikace navržená tímto způsobem je teoreticky „neprůstřelná“ (i do budoucna) a její zobrazení bude na všech zařízeních vždy optimální. Bohužel ale vše má svou cenu. Doba návrhu a implementace takové aplikace nemusí vždy korespondovat s dobou návrhu stránky dle jednoho z jednodušších způsobů jmenovaných dříve.

Výsledek tohoto experimentálního přístupu k aplikaci:

- ✓ Různý obsah pro různá zařízení
 - ✓ Rozeznáváme pomocí hodnot prohlížeče, rozlišení displeje i DPI
 - ✓ Správné zobrazení obrázků na zařízeních s vysokým DPI
 - ✓ Funguje i na starších platformách
- × Velmi složitá implementace

4.8 SROVNÁNÍ PŘÍSTUPŮ

Jaké je tedy naše východisko. Je vhodné optimalizovat čistě na bázi rozlišení nebo je dobré se zaměřovat komplexně i na DPI? Jak už to tak bývá, nejlepší je stát někde mezi oběma těmito možnostmi a soustředit se na jejich vyvážení.

V současnosti by stačilo, kdyby byl webový designer schopný implementovat i tu nejjednodušší možnost jmenovanou v prvním příkladě. Pro nejlepší výsledek je sice nejlepší zakládat webovou aplikaci na komplexním řešení DPI i rozlišení, ale pro rozumnou dobu vývoje aplikace si vystačíme s její jednodušší verzí v podobě viewportu.

Nejlepší možností je tedy nastavit v metadatech náležitě `viewport`, čímž docílíme jisté unifikace vizualizace v rámci DPI a dále se chovat k webové aplikaci na bázi rozlišení displeje zařízení (nebo jeho WWW prohlížeče) měněním CSS stylů HTML elementů. Nedocílíme tak sice za každých okolností absolutně příznivých výsledků, ale pro desktopová a mobilní zařízení současnosti nám tento způsob bohatě postačuje.

5 ZHODNOCENÍ VLASTNOSTÍ ZAŘÍZENÍ

V této etapě diplomové práce již nalezneme rozřešení optimalizace vlastností zařízení. Zdroje problémů a možná východiska jsem rozdělil dle kategorizace, kterou jsem vytvořil na bázi analýzy možných příčin. Nejdříve se budu věnovat problémům v rámci hardwaru, poté přejdu na softwarové nedokonalosti a nakonec se zaměřím na možné problémy z hlediska dat.

5.1 HARDWARE

Problémy spojené s hardwarem uživatel WWW aplikace nijak nemůže ovlivnit. Jsou dané vlastnostmi zařízení, které mají svůj počátek u výrobce dané platformy. Naštěstí my, jakožto weboví designeři, jsme schopni problémům tohoto typu předejít.

5.1.1 Rozlišení a DPI

Možná řešení v rámci rozlišení a DPI jsem již rozebral ve specifikaci přístupů k webovým aplikacím. Nyní si však ukážeme jedno z optimálních řešení tohoto problému, které jsem použil v praktické části této práce.

Zprvu nastavíme v metadatech `viewport` a poté vytvoříme Media Queries pro aplikaci CSS stylů pro různá zařízení.

```
// HTML - Metadata Viewport

<meta name="viewport"
      content="width=device-width; initial-scale=1.0;"
/>

// CSS - Media Queries

@media screen and (max-width: 600px) {
  /* CSS pro mobilní zařízení */
  ...
}
@media screen and (min-width: 601px) {
  /* CSS pro desktop */
  ...
}
```

Obrázek 31. Rozlišení a DPI

5.1.2 Operační paměť

Ošetřit problémy v rámci operační paměti (RAM) můžeme pouze domluvou a jinak je nemůžeme o moc blíže specifikovat, jelikož pomocí technologií pro vývoj webových aplikací nemůžeme zjistit, jak velkou operační paměť zařízení má. Avšak i tak je obecně potřeba brát ohled na (zejména mobilní) zařízení, která mají velmi malou operační paměť a uzpůsobit tomu tak naši WWW aplikaci.

To znamená např., že nesmíme používat příliš velké obrázky, které by se musely poté zmenšovat pomocí CSS např. u náhledů. WWW prohlížeč by tak zbytečně načítal

obrázky s větší datovou velikostí, než kolik by bylo potřeba, kdyby byl obrázek takovému náhledu uzpůsoben. Pro takové případy je dobré mít 2 varianty daného obrázku.

Dále je dobrým zvykem nemít na jedné stránce obrázků příliš. A to nejen z hlediska operační paměti. Problém pak nastává i s rychlostí načítání takové WWW aplikace (k čemuž se ještě dostaneme).

Není špatnou zvyklostí před implementací nějaké JavaScriptové funkce testovat její funkčnost. A to zejména tedy vnořené smyčky, které mohou být příčinou závažných problémů.

V neposlední řadě je obecně dobré odebrat u WWW stránek pro mobilní zařízení (nebo obecně pro zařízení s malou kapacitou RAM) nepotřebný obsah a vytvořit obsah krátký a výstižný.

5.1.3 Ovládání

Zařízení současnosti můžeme v základu dělit dle způsobu ovládání, jak vyplynulo z analýzy, na dva typy:

- ❖ Zařízení bez dotykového ovládání
- ❖ Zařízení s dotykovým ovládáním

Interaktivní grafické prohlížeče mohou měnit způsob vizualizace určitých elementů na základě chování uživatele na webové stránce. Přesto ale ne všechny způsoby lze vždy využít při realizaci takové aplikace.

Při vývoji stránek pro zařízení bez dotykového ovládání se nemusíme nijak omezovat, nicméně pouze v případě nepočítáme-li do této kategorie bezdotykové mobilní telefony nebo ostatní zařízení bez myši nebo trackballu (tato zařízení můžeme bez ostychu zařadit do kategorie druhé). Uživatel pro zařízení v této kategorii může využít veskrze všechny možnosti ovládání, které navrhne.

Oproti tomu má-li uživatel zařízení uzpůsobené k dotykovému ovládání, již nastává komplikace s elementy, které reagují na „njetí nad element“. V takovém případě nesmíme při vytváření našich ovládacích prvků využívat během stylizace elementů pseudotřidu `:hover`.

```
// CSS stylizace pro odkazy - přípustné pro dotykové ovládání

A:link          /* Odkaz */
A:visited       /* Navštívený odkaz */
A:active        /* Aktivovaný odkaz (např. stisknutí tlačítka) */
A:focus        /* Aktivní odkaz (např. aktivní input element) */

// CSS stylizace pro odkazy - nepřípustné pro dotykové ovládání

A:hover         /* Odkaz, nad kterým stojíme s kurzorem */
```

Obrázek 32. CSS stylizace odkazů

Tento problém se nezdá být až tak závažný, jeho relevantnost se ovšem projeví až v kombinaci s JavaScriptem (nebo jQuery), kdy jsou pomocí těchto CSS stylů tvořeny např. důležité ovládací prvky, pomocí kterých se pohybujeme po stránce. Bude-li tedy taková WWW aplikace mít hlavní menu viditelné až po „njetí nad element“, tak jej uživatel na dotykovém zařízení nebude moci zobrazit.

5.2 SOFTWARE

Zdroje problémů v této kategorii závisí pouze na uživateli a jeho výběru použitého softwaru. Jedná se o dynamicky rozdílné zdroje problémů, které mohou být i u zařízení stejného typu odlišné. Zda bude mít koncový uživatel problémy s těmito zdroji problémů, závisí pouze na webovém designerovi a jeho zkušenostech s tvorbou WWW aplikací.

5.2.1 Webový prohlížeč

WWW prohlížeče můžeme v základu rozdělit do dvou kategorií, dle způsobu vizualizace webového obsahu:

- ❖ Textový prohlížeč
- ❖ Grafický prohlížeč

Problém s textovým prohlížečem může nastat, jakmile se na webové stránce zobrazí příliš mnoho obrázků, které s sebou nesou významný informační obsah.

Předejít těmto problémům můžeme např., vynecháme-li takové obrázky s důležitou informační hodnotou. Takový způsob řešení ovšem není příliš vhodný, protože používání čistě textových prohlížečů je v současnosti v úpadku a nevyužili bychom tak potenciál grafických prohlížečů nynější doby.

Druhý, lepší způsob, tkví v řádném a vystihujícím popisu obrázků (``), obrázkových map (`<area>`) a obrázkových tlačítek (`<input type="image">`). Takový způsob je vhodný nejen pro zobrazování webového obsahu v textových prohlížečích, ale např. i pro jeho zpracování hlasovými čtečkami.

HTML elementy vyjadřující obrázek tedy opatřujeme atributem `alt`, do kterého vepíšeme popis obrázku. Jestliže význam obrázku nelze dostatečně stručně popsat, můžeme použít atribut `longdesc` a opatřit ho odkazem na WWW stránku s detailním popisem. Obvykle tedy v dnešní době pomocí tohoto atributu odkazujeme na internetovou encyklopedii Wikipedia.

```
  

```

Obrázek 33. Popis obrázkových elementů

Obecně uzpůsobení webového obsahu pro WWW prohlížeče tkví zejména ve správné struktuře HTML dokumentu. Nesmíme zapomenout na úvodní `DOCTYPE` dokumentu, správné popsání metadat a bezchybně používat HTML elementy. Takový HTML dokument se považuje za validní.

5.2.2 Webový prohlížeč – HTML5

Jak jsem již zmiňoval, tak v praktické části této práce jsem pro vývoj webové aplikace použil novou rozvíjející se technologii HTML5. Nynější webové prohlížeče se zobrazením obsahu, strukturovaném pomocí této technologie, problém nemají. Potíž nastává až u prohlížečů starších, které elementy, uvedené až s HTML5 specifikací, nedokáží rozpoznat a náležitě je tedy neumí strukturovat a stylizovat. V takovém případě se na starších WWW prohlížečích musíme spoléhat na přítomnost JavaScriptu a pomocí něj problém ošetřit vytvořením DOM objektů pro neznámé elementy.

```
// Skript vyvoláme pouze pro starší prohlížeče

<!--[if lt IE 9]>

    // Vytvoření HTML5 elementu article pomocí JavaScriptu

    <script language="JavaScript" type="text/javascript">
        document.createElement("article");
    </script>

<![endif]-->
```

Obrázek 34. Demonstrace zpřístupnění HTML5 pomocí JavaScriptu

5.2.3 Webový prohlížeč – CSS3

Technologii CSS3 bohužel nemůžeme zpřístupnit všem prohlížečům jako je tomu s HTML5. Chceme-li vytvářet webový obsah stylizovatelný i staršími grafickými prohlížeči, nesmíme buď CSS3 použít vůbec nebo musíme pro elementy stylizované těmito kaskádovými styly vytvořit i alternativní styl pomocí CSS2 či CSS1. I když tak nebude webový obsah ve všech prohlížečích identicky stylizovaný, umožníme tak starším prohlížečům jej alespoň zobrazit podobným způsobem.

Řešení takového problému můžeme implementovat prakticky obdobným způsobem, jako tomu bylo v předchozím případě, a to na bázi podmíněných komentářů.

```
<link rel="stylesheet" type="text/css" href="CSS3.css">

<!--[if lt IE 9]>
    <link rel="stylesheet" type="text/css" href="CSS1a2.css">
<![endif]-->
```

Obrázek 35. Optimalizace CSS3

5.2.4 Webový prohlížeč – AJAX

Ve zkratce bychom mohli říci, že webová technologie AJAX funguje ve všech prohlížečích, které podporují JavaScript. Ovšem ne vždy je tomu tak. Zejména tedy mobilní telefony vyžadují JavaScriptovou část AJAXového volání přístupnou v samotném HTML dokumentu, kde AJAX používáme, a nelze ji oddělit do samostatného souboru. Je potřeba na tento fakt brát ohled a uzpůsobit tímto stylem mobilní stránky aplikace všude tam, kde budeme dynamický obsah načítat.

5.2.5 Moduly

Volitelné moduly typu Flash, Silverlight a další nejsou vhodné pro tvorbu široce přístupných aplikací. Naznačuje tomu i fakt, že jejich existence je ze světa internetu poslední dobou značně vytlačována a také to, že nová mobilní zařízení již v základu např. Flash podporovat nebudou. Vskrze vše, co lze vytvořit pomocí těchto technologií, které vyžadují nějaký tento rozšiřující modul, lze vytvořit pomocí technologií HTML5, CSS(3) a JavaScript.

Pokud chceme na webově aplikaci prezentovat nějaký multimediální obsah, např. video, které je v současnosti velmi oblíbené implementovat formou objektu, vytvořené v modulu Flash, poslouží nám element `<video>`, kterým jej na stránku můžeme umístit a náležitě s ním pracovat.

Animace nebo další interaktivní části webové aplikace můžeme též vytvořit pomocí tří výše zmíněných technologií.

Dovolím si říct, že jejich existence je ve webovém prostředí nepotřebná a formě „otevřeného webu“ nevyhovující.

5.3 DATA

Webové stránky si prohlížíme s jediným úmyslem, vždy chceme získat nějaká data. Data z internetu potřebujeme „vydolovat“ co nejrychleji a pokud možno za co nejnižší cenu. Proto se naskytá otázka, jak optimalizovat naši webovou aplikaci pro tyto dva faktory.

5.3.1 Rychlost dat

Rychlost načítání webové aplikace závisí na mnoha faktorech, přičemž některé ovlivnit můžeme a některé bohužel ne:

- ❖ Datová velikost WWW stránky
- ❖ Struktura databázových (SQL) dotazů
- ❖ JavaScriptové funkce
- ❖ Výběr poskytovatele webového prostoru
- ❖ ISP uživatele

Webovou aplikaci musíme zejména uzpůsobit tak, aby byl její obsah v rámci unesitelné datové velikosti. Správně strukturovat její dynamické odkazy mezi více menších HTML stránek s rozumným množstvím multimediálního obsahu.

Konstrukce dotazů, kterými vybíráme data z databáze, musí být optimální a nikterak nepřevyšující složitost dané situace. Je vždy výhodnější nejprve sloučit (JOIN) např. dvě tabulky, ze kterých potřebujeme data naráz, než volat dva oddělené dotazy.

JavaScriptové funkce nám mohou drasticky zpomalit zobrazení WWW stránky na pomalejších platformách. Je potřeba brát ohled na jejich strukturu a nečinit je zbytečně obsáhlé, jelikož zpracování JavaScriptových částí HTML dokumentu má na starost webový prohlížeč koncového zařízení uživatele.

Výběr poskytovatele webového prostoru je neméně důležitý faktor, nad kterým je nutné se pozastavit. Chceme-li aby byla naše webová aplikace „svižná“, je vždy lepší vybrat si nějakého placeného poskytovatele v rámci státu.

ISP uživatele (Internet Service Provider; česky poskytoval připojení k internetu) bohužel neovlivníme. Na tento faktor je důležité se zaměřit hlavně vzhledem k datové velikosti aplikace, kdy uživatel nemusí mít vždy optimálně rychlé připojení.

5.3.2 Cena dat

V současnosti, kdy je prakticky na každém rohu WiFi AP zdarma souběžně s koexistencí dedikovaných připojení pro desktopová zařízení, by se mohl tento problém jevit jako triviální. Pořád ale musíme brát v úvahu, že uživatel s mobilním zařízením může k webové aplikaci přistupovat skrze roaming, kdy platí závisle na množství přenesených dat nebo má paušál, kdy nechce překročit svůj datový „strop“. Proto je potřeba brát ohled i na tyto možnosti a WWW aplikaci pro mobilní zařízení tomuto problému náležitě datově uzpůsobit např. menší velikostí HTML dokumentů nebo zpřístupněním čistě textové podoby aplikace.

6 NÁVRH WEBOVÉ APLIKACE

Po zhodnocení vlastností zařízení a problémů s nimi spojených, u kterých jsme zároveň naplánovali jejich ošetření, se můžeme zaměřit na praktickou část této diplomové práce a začít s těmito nabytými znalostmi operovat v rámci tvoření webové aplikace.

6.1 NÁVRH ARCHITEKTURY SYSTÉMU

Architektura systému je obecný popis charakteru aplikace. Její význam tkví ve vymezení základních komponent a vztahů mezi nimi.

Zprvu tedy objasním cíle vizuálního rozvržení portálu, které jsem si vztyčil, a následně navrhnu samotnou architekturu aplikace.

6.1.1 Volba Internet Information Services

I když jsem původně vyvíjel portál pod architekturou AMP (Apache, MySQL a PHP), tak jsem kvůli zakomponování technologie Shibboleth nakonec zvolil webový server IIS.

Důvod pro volbu IIS je čistě praktický, neexistuje v současnosti žádný oficiální ani neoficiální informační zdroj pro zakomponování technologie Shibboleth do modulu AMP pod Windows 7, přičemž pro IIS 7.5 ano [16].

6.1.2 Cíl vizuálního rozvržení

Před samotnou tvorbou aplikace jsem přemýšlel nad tím, jaké potřeby mají uživatelé přistupující ke komunikačnímu portálu z různých zařízení. Když jsem hodnotil vlastnosti zařízení, tak jsem příliš nedbal na lidský faktor, ale nyní přichází vhodný čas pro jeho zahrnutí.

Uživatelé, kteří k portálu přistupují z desktopových zařízení (zařadíme do této kategorie i tablety pro jejich hardwarovou podobnost), obvykle vyžadují na své obrazovce přehled. Chtějí mít všechna důležitá oznámení, novinky a události na jednom viditelném místě.

Oproti tomu uživatelé přistupující z mobilních zařízení mají potřebu pomoci jednoho kliknutí zobrazit jimi vyžadovaný obsah co nejrychleji, i když se na jejich (obvykle malý) displej nemohou všechny informace vtěsnat najednou. Z tohoto důvodu je u studentského komunikačního portálu přístupnému skrze mobilní zařízení vhodné zvolit jeho ovládání formou hlavního menu a následně se na důležité podstránky odkazovat.

Jako cíl při tvorbě WWW aplikace jsem si tedy proto vytyčil formu dvou vlastních pod-portálů, které ovšem sdílí určité zdroje, jež jsou pro jejich společný souběh typické. Samozřejmostí je i sdílení stejné databáze aby mohla probíhat komunikace mezi uživateli připojenými z obou druhů zařízení.

6.1.3 Proč volba dvou souběžných pod-portálů?

Předpokládal jsem zcela odlišnou strukturu vizuálního rozvržení studentského komunikačního portálu na desktopovém a mobilním zařízení. Z tohoto důvodu jsem nezvolil jedinou adaptivní aplikaci, kde by se měnil pouze vzhled, ale vytvořil jsem dva pod-portály, kdy jeden je určen pro desktopová zařízení a druhý pro zařízení mobilní.

Uživatelé na mobilních zařízeních očekávají od tohoto typu webové aplikace zcela jinou zkušenost než uživatelé na zařízeních desktopových. Uživatelské rozhraní je tak v tomto smyslu uzpůsobeno oběma platformám odděleně.

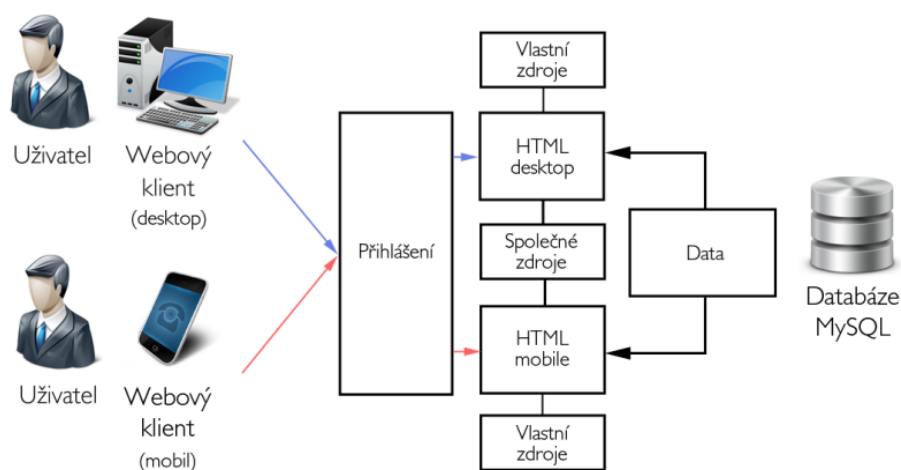
Na mobilních zařízeních uživatelé vyžadují velmi jednoduché ovládání, tudíž jsem zvolil jako ústřední ovládací prvek hlavní menu, ze kterého může uživatel rychle přejít do jakékoli sekce komunikačního portálu.

Uživatelé na zařízeních desktopových potřebují mít co nejvíc informací viditelných po celou dobu strávenou na komunikačním portálu. Uživatelské rozhraní jsem tedy rozdělil na tři sloupce, kdy v levém jsou základní informace a hlavní menu, prostřední sloupec se dynamicky mění v závislosti na prohlížené sekci a sloupec vpravo ukazuje dodatečné informace.

Z důvodu takto rozdílných požadavků na uživatelské rozhraní jsem zvolil tvorbu dvou souběžných pod-portálů.

6.1.4 Architektura systému

Architekturu systému jsem tedy přizpůsobil svému cíli a uskutečnil dle následující vizuální reprezentace.



Obrázek 36. Architektura systému

Uživatele po přihlášení do systému v závislosti na jeho zařízení přesuneme na pod-portál jemu určený. Tuto skutečnost vyjadřují různě barevné šipky, které znázorňují různé druhy přístupu. Po přestupu na další stránku je uživatelův pohyb ustálen a stránky, na které se nachází, jsou zpřístupněny zdroje jí vlastní a zároveň i zdroje, které jsou pro oba tyto pod-portály společné. Jelikož komunikace probíhá přes databázi, tak je její obsah společný také pro obě části portálu.

6.2 NÁVRH STRUKTURY PORTÁLU

Po návrhu struktury architektury a přiblížení její ideje se můžeme zaměřit na podrobnější strukturu portálu. Navržení a znázornění takové struktury je důležité pro bližší pochopení funkčnosti celé aplikace, jelikož se v tomto případě nejedná o triviální problém.

Zprvu v této etapě tedy vyložím cíle struktury celého portálu, následně ji pak graficky vyobrazím a popíši průběh chování této aplikace.

6.2.1 Studentský komunikační portál

Jako aplikaci pro demonstraci přístupu k WWW stránce z různých typů zařízení jsem zvolil studentský komunikační portál. Ten v základu umožňuje studentům vzájemnou výměnu zpráv mezi studenty a jejich skupinami.

Studenti mohou vytvářet (či mazat) vlastní skupiny a přidávat do nich své přátele, sledovat kolik lidí mají ve skupinách ostatní uživatelé, mohou si mezi sebou posílat soukromé zprávy, hledat na portálu další uživatele, upravovat si vlastní profilové informace, sledovat důležité události generované systémem, prohlížet si základní informace ostatních uživatelů, reagovat na napsané zprávy formou komentářů, sledovat návštěvy na své profilové stránce i na cizích profilových stránkách nebo sledovat aktivitu přátel.

Nad společnou základnou, která obsahuje zejména data a všechny tyto funkce, vznikly pod-portály dva – jeden pro desktopové a druhý pro mobilní klienty.

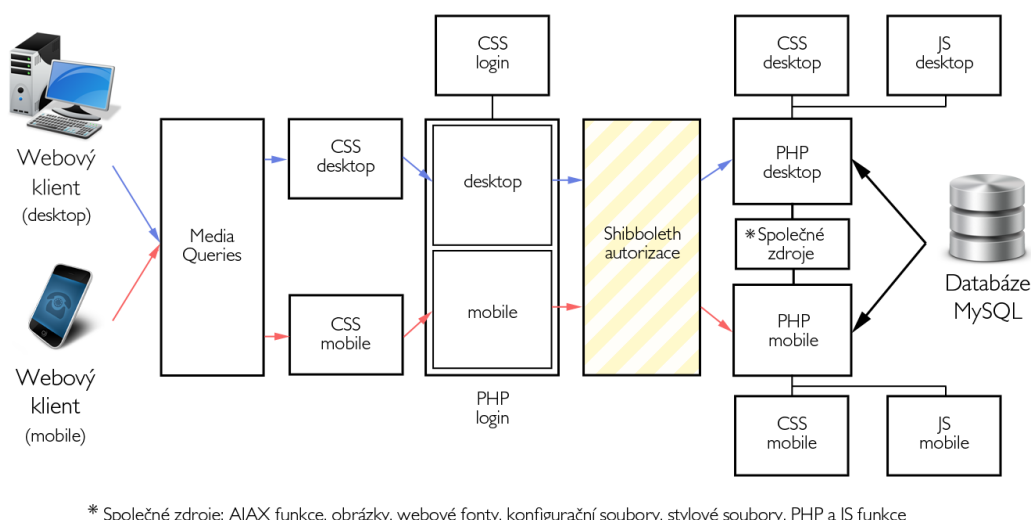
6.2.2 Cíl struktury komunikačního portálu

Základní myšlenka strukturace tohoto komunikačního portálu vyvstala, ač nepřímou, již během analýzy a rozboru. Struktura portálu je založena na modelu z kapitoly 4.3 *Media Queries – Rozcestník* a je rozšířena o další části aplikace.

Jedním z hlavních cílů je kategorizace zařízení a přizpůsobení vizualizace webové aplikace. Následně pak autentizace uživatelů pomocí technologie Shibboleth. Dalším cílem je sdílení společných zdrojů, jako jsou např. obrázky a funkce. Neposledním cílem je oddělit funkčnost zdrojů nesdílených, aby se nenarušoval souběh pod-portálů. V závěru je podstatné pro obě části aplikace vytvořit stejné databázové spojení a mít tak nadosah stejná data.

6.2.3 Struktura portálu a Shibboleth

Strukturu portálu nám nejlépe vystihne následující obrázek. Pro snadnější vyjádření a pochopení celé struktury jsem vypustil některé méně podstatné části.



Obrázek 37. Obecná struktura portálu a Shibboleth

S nastaveným meta tagem `viewport` v hlavičce dokumentu můžeme operovat s webovou aplikací na základě Media Queries a rozlišení zařízení.

Dle rozlišení jsem tedy pro různá zařízení nastavil rozdílnou viditelnost HTML elementů v PHP dokumentu vyhrazeném pro přihlašování uživatelů a zobrazil tak požadovaný obsah, který je náležitě stylizován pomocí CSS určeném tomuto dokumentu.

Samotné přihlášení uživatele probíhá skrze systém Shibboleth, kdy je IdP (identity provider) poslána různá návratová URI adresa v závislosti na zařízení, které uživatel pro přístup k aplikaci používá.

Následně se uživatel ocitne na jemu určeném pod-portálu, který je opatřen svými privátními CSS a JS soubory. Oběma pod-portálům ovšem přiděluji i společné zdroje jako např. AJAX funkce, obrázky nebo webové fonty.

Umožnění komunikace uživatelů nezávisle na platformě zajišťuji skrze stejnou databázi pro oba pod-portály.

6.3 NÁVRH STRUKTURY DATABÁZE

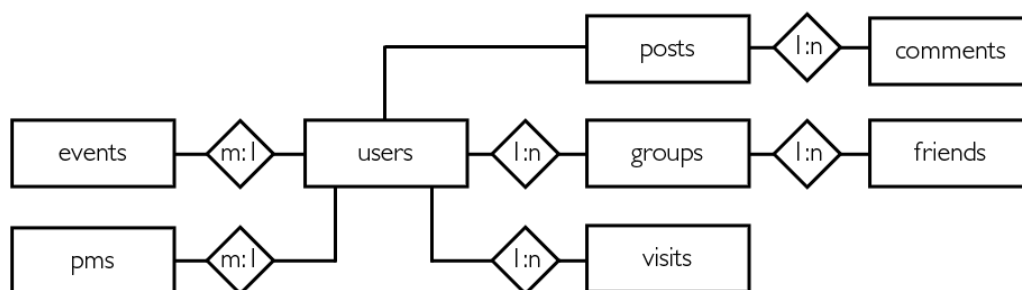
Zvolil jsem databázi MySQL a to zejména na základě vlastních zkušeností, ale také kvůli její multi-platformnosti. Výhodou při výběru bylo také to, že se jedná o open source projekt, jelikož jsem zastánce „open“ technologií.

6.3.1 Cíl struktury databáze

Jedná se o bázi všech dat, které se zrcadlí mezi oba pod-portály, proto jsem si za cíl kladl její použitelnost na obou pod-portálech.

6.3.2 Struktura databáze

Jelikož není tvorba databáze stěžejní částí této diplomové práce, tak vypustím popis analýzy a návrhu databáze a uvedu pouze výsledek.



Obrázek 38. Struktura databáze – tabulky a relace

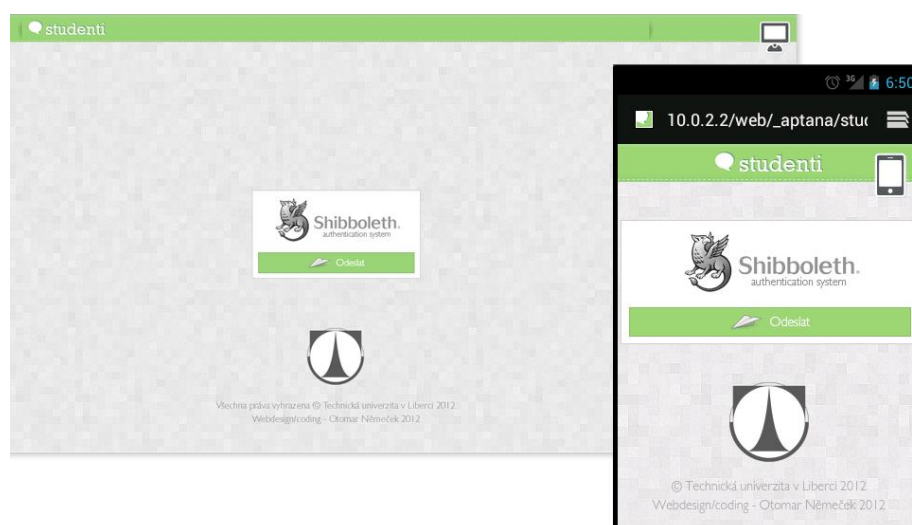
Tabulka `users` obsahuje veškerá data o uživatelích, včetně jejich aktivity v různých částech aplikace. V tabulce `visits` uchovávám návštěvy profilových stránek, což jsou osobní stránky uživatelů, které o nich obsahují základní informace. Tabulka `events` zahrnuje veškeré automaticky generované události, o kterých by měl uživatel vědět jako např., že byl přidán do nějaké skupiny. V tabulce `pms` jsou soukromé zprávy mezi uživateli. Tabulka `groups` obsahuje skupiny vytvořené uživateli a tabulka `friends` přátele, kteří do nich patří. V tabulce `posts` jsou veškeré zprávy posílané veřejně nebo uzavřeným skupinám a v tabulce `comments` jsou jejich komentáře.

7 TVORBA APLIKAČNÍCH STRÁNEK

V této závěrečné kapitole se budu věnovat tvorbě jednotlivých částí studentského komunikačního portálu, jejich funkčním činnostem, uživatelskému rozhraní a rozdílům přístupu z různých zařízení.

Již v návrhu jsem zmínil, že jsem vyvinul dva souběžné pod-portály pro mobilní a desktopová zařízení. Při popisu uživatelského rozhraní ukáži vždy, jak vypadá verze dané části pro obě tyto platformy.

7.1 PŘIHLÁŠENÍ



Obrázek 39. Přihlášení

Přihlašovací stránka se skládá ze samotného formuláře pro přechod na autentizační aplikaci Shibboleth a z tlačítka (vpravo nahoře), které znázorňuje zařízení, na kterém vizualizujeme WWW aplikaci. Záměrně říkám tlačítka, protože pomocí něj můžeme mezi oběma módy přepínat a procházet webovou aplikaci v obou podobách na všech zařízeních, budeme-li si to přát. Zároveň jsme na této stránce obeznámeni i s autorskými právy, která patří Technické univerzitě v Liberci a s autorem této aplikace – mou maličkostí.

7.1.1 Princip přihlášení

Přihlášení je první část portálu, se kterou uživatel konfrontuje během jeho návštěvy portálu. Způsob přihlášení tkví v přesměrování WWW klienta uživatele na IdP (identity provider) Technické univerzity v Liberci, odkud je pak po vyplnění formuláře (emailová adresa TUL a heslo) odkázán zpět na komunikační portál, kde následně zpracovávám obdržená data.

Po přihlášení je uživateli vytvořeno sezení (`$_SESSION["id"]`), přes které s jeho aktivitou na portálu dále náležitě operuji. Pro samotnou práci s id sezení jsem vytvořil proměnnou `$sid`.

7.1.2 Detekce zařízení

Před samotnou detekcí jsem v hlavičce dokumentu nastavil `viewport`, abych unifikoval rozlišení na zařízeních s vysokým rozlišením a DPI. S nastavenou šířkou viewportu na `device-width` se moderní (mobilní) zařízení s vysokým rozlišením a DPI navenek tváří jakoby měla rozlišení se šířkou 320 pixelů. Tudíž pak lze provádět detekci zařízení pouze na bázi šířky displeje a prohlížeče.

```
<meta name="viewport" content="width=device-width; initial-scale=1.0">
```

Obrázek 40. Nastavení meta tagu viewport

Po nastaveném viewportu jsem již řešil samotnou detekci pomocí Media Queries. Rozhodl jsem, že zařízení se šířkou menší než 600 pixelů budu považovat za mobilní a ta, která mají šířku větší, budu považovat za desktopová. U mobilních zařízení se rozměry určují při držení v „portrait“ módu, což znamená držení na výšku.

Přihlašovací stránka je rozdělena na dva bloky, přičemž viditelný je vždy pouze jeden. První blok je určený pro mobilní zařízení a druhý pro desktopová. V každém bloku je i různý přihlašovací formulář. Každý formulář obsahuje různou návratovou adresu, kterou posíláme IdP Shibboleth (TUL).

Viditelnost bloků určuji právě na bázi zařízení, které tuto přihlašovací stránku zobrazuje. Po detekci zařízení pomocí Media Queries se skrze CSS zakáže vizualizace bloku, který je určen pro jiná zařízení.

```
/* Mobilní zařízení */
@media screen and (max-device-width: 600px) {
  #full-container {
    display: none;
  } }

/* Desktopová zařízení */
@media screen and (min-device-width: 601px) {
  #full-container-mobile {
    display: none;
  } }
```

Obrázek 41. Nastavení Media Queries a zobrazování bloků

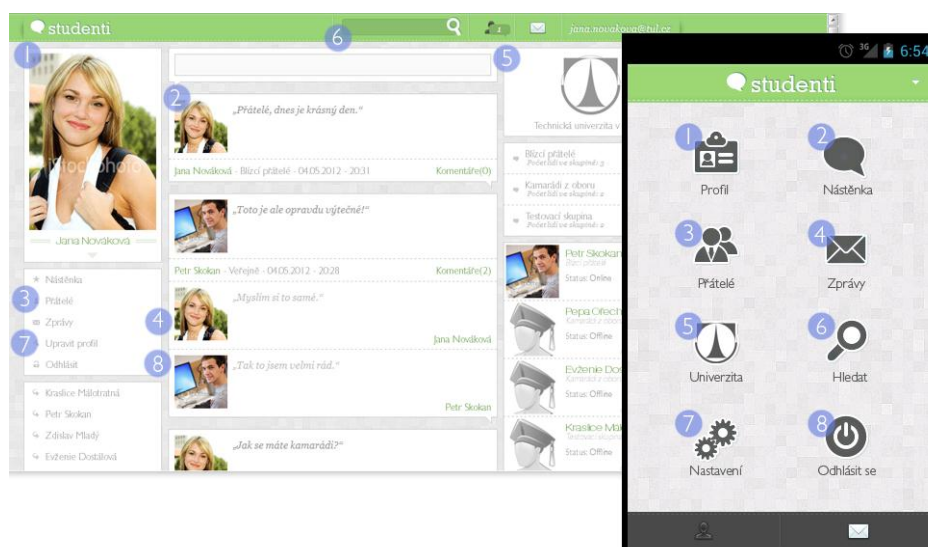
7.1.3 Rozdíly přístupu

Rozdíly mezi přihlašovací stránkou pro desktopová a mobilní zařízení tkví zejména v přihlašovacích formulářích.

Formulář, určený pro desktopová zařízení, se vystředí na obrazovku a obsahuje návratovou adresu pro IdP Shibboleth, skrze kterou bude poté uživatel přesměrován na pod-portál určený pro desktopová zařízení.

Formulář, určený pro mobilní zařízení, je zamýšlen pro menší displeje, proto se roztáhne na celou šířku. Obsahuje návratovou adresu pro IdP Shibboleth, skrze kterou bude poté uživatel přesměrován na pod-portál určený pro zařízení mobilní.

7.2 UŽIVATELSKÉ ROZHRAŇÍ



Obrázek 42. Uživatelské rozhraní

Jak z vizuální reprezentace vyplývá, uživatelské rozhraní je sice jinak strukturované, ale nabízí uživatelům na všech platformách úplně stejné možnosti. V obou případech tedy máme přístupný profil (1), ovšem na mobilním zařízení jej musíme nejdříve zobrazit. Dále je přístupná nástěnka (2), která je na desktopové verzi portálu viditelná ihned po samotném přihlášení uživatele. Tlačítka přátelé (3) a zprávy (4) jsou dalšími součástmi rozhraní. V obou verzích portálu dále můžeme najít odkaz (5) na domácí stránku Technické univerzity v Liberci. Nechybí ani modul hledání (6) a nastavení profilu (7). Esenciální součástí je samozřejmě i odhlášení (8) z aplikace.

V desktopové verzi portálu vidíme upozornění na nové zprávy a události v horní liště portálu, přičemž v mobilní verzi byla tato upozornění pro přehlednost a kvůli nedostatku místa na malých displejích přesunuta do spodní části. Samotná upozornění jsou aktivní oblasti, na které když klikneme, budeme přesunuti na požadovanou stránku.

Po kliknutí na svou emailovou adresu v horní liště se nám zobrazí vysouvací menu, ve kterém jsou obsaženy odkazy pro rychlou navigaci – profil, nastavení profilu a odhlášení. Na mobilní verzi portálu je toto menu také přístupné na stejném místě, ale skrývá se pod šipkou.

7.2.1 Uživatelské rozhraní na mobilních zařízeních

Ústřední prvek uživatelského rozhraní na mobilních zařízeních je, jak vyplývá z předchozího obrázku, hlavní menu. To je zobrazeno bezprostředně po přihlášení uživatele. Pomocí tohoto menu může uživatel přejít na jakoukoli část pod-portálu, který je určen mobilním zařízením. Z každé části se pak může tlačítkem „Zpět“, které je vždy vyobrazeno v levé části horní lišty, vrátit zpět na hlavní menu.



Obrázek 43. Tlačítko Zpět

Ovládání formou hlavního menu jsem zvolil kvůli uchování jednoduchosti aplikace. Uživatel, přistupující ke studentskému portálu z mobilního zařízení, je tak pouze obeznámen s částmi portálu, na které může (a nemusí) přejít.

Hlavní menu je z hlediska dat velmi nenáročné. Na této stránce nevolám žádné MySQL dotazy, kromě těch, které vyžaduje spodní lišta, kde se kontrolují nové zprávy a události.

7.2.2 Uživatelské rozhraní na desktopových zařízeních

Uživatelské rozhraní na desktopových zařízeních je rozděleno na tři části, resp. sloupce.

Levý sloupec vždy ukazuje fotografii, na kterou když najedeme myší, zobrazí se základní informace o uživateli. Informace můžeme zobrazit i kliknutím na šipku, která je umístěna ve spodní části. Pod fotografií je ovládací menu, kterým můžeme přejít na nástěnku, zobrazit přátele či zprávy, upravit profil nebo se odhlásit. Pod tímto menu je zobrazeno pět posledních návštěv uživatele.

Prostřední sloupec se dynamicky mění v závislosti na prohlížené stránce. Bezprostředně po přihlášení je uživateli zobrazena jeho nástěnka, která obsahuje různé zprávy od jeho přátel.

Pravý sloupec zobrazuje logo Technické univerzity v Liberci, které když stiskneme, budeme přesunuti na domovskou stránku TUL. Pod tímto logem jsou zobrazeny skupiny uživatele a počet lidí, které v dané skupině má. Pod skupinami je zobrazena aktivita pěti nejaktivnějších přátel uživatele s časem jejich poslední činnosti.

7.2.3 Nové zprávy a události

Upozorňování na nové zprávy a události je řešeno formou zobrazení „bublinky“ s počtem nových zpráv či událostí. Na desktopových zařízeních jsou tato upozornění umístěna na horní liště, přičemž na mobilních zařízeních jsou tato upozornění na liště spodní.



Obrázek 44. Upozornění na mobilních zařízeních – Spodní lišta



Obrázek 45. Upozornění na desktopových zařízeních – Horní lišta

Principiálně ovšem zjišťování nových zpráv a událostí funguje na obou portálech stejně. Upozornění na nové události (či zprávy) je tvořeno pomocí Unixových časových stop, kdy se porovnává momentální čas a čas posledního zobrazení rozhraní událostí z čehož se pak počítá počet událostí, které uživatel ještě nezobrazil. Následující kód znázorňuje funkčnost zjišťování nových událostí. Zjišťování nových zpráv funguje obdobným způsobem.

```

<?php
$db_call = mysql_query("select count(*) as pocet from events
                        where cas > $sid_udalosti and
                        uid = $sid;");
$db = MySQL_Fetch_Array($db_call);
$pocet = $db["pocet"];
?>

<div class="f-td">
  <div class="footer-udalosti-bubble">
    <a href="m-udalosti.php">
      <?php echo $pocet;?>
    </a>
  </div>
  <a href="m-udalosti.php">
    
  </a>
</div>
<?php
if($pocet > 0){
echo "<script language='JavaScript' type='text/javascript'"
    $(''.footer-udalosti-bubble').slideToggle(100);
    </script>";
}
?>

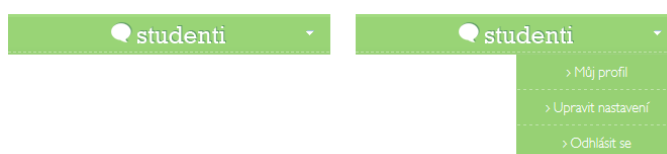
```

Obrázek 46. Upozornění na nové události – PHP, HTML, MySQL a jQuery

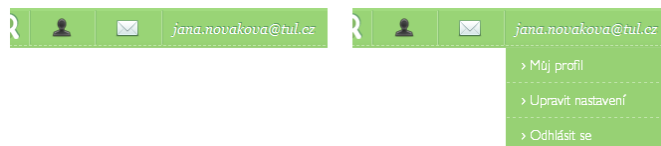
7.2.4 Vysouvací menu

Horní lišta na desktopovém zařízení a horní i spodní lišta na mobilním zařízení jsou stále viditelné na všech stránkách pod-portálů. Chtěl jsem, aby bylo vysouvací menu stále přístupné, protože nabízí možnosti, které uživatel může za určitých okolností akutně potřebovat. Z tohoto důvodu jsem jej v obou verzích pod-portálů umístil do pravé části horní lišty.

Vysouvací menu obsahuje možnosti pro přechod na profilovou stránku uživatele, nastavení profilové stránky a odhlášení z portálu.



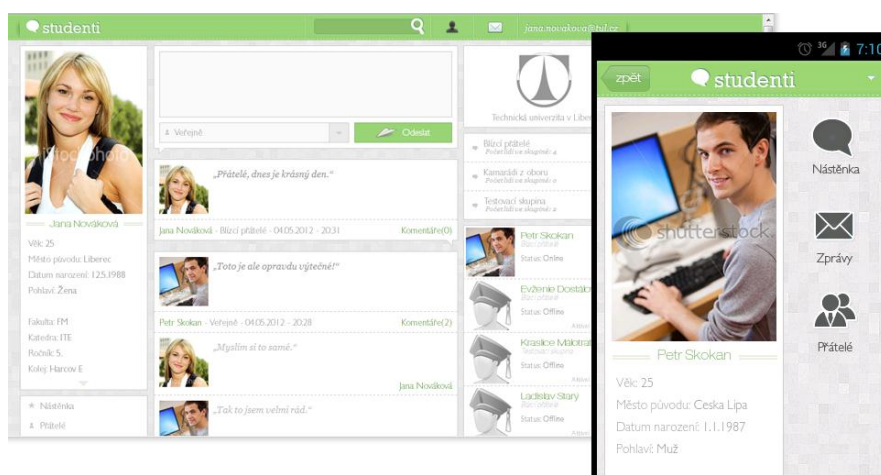
Obrázek 47. Vysouvací menu – mobilní verze



Obrázek 48. Vysouvací menu – desktopová verze

Toto vysouvací menu se zobrazí po kliknutí na šipku (mobilní verze) nebo email (desktopová verze). Pro tvorbu odkazů na stránky, na které přecházíme skrze toto vysouvací menu, jsem nepoužil žádná data z MySQL databáze. Rozlišení uživatelů na stránkách, kam nás odkazuje toto menu, jsem založil na ID sezení (\$_SESSION["id"]).

7.3 PROFIL



Obrázek 49. Profil

Profil je osobní stránka uživatele (mobilní verze) nebo část stránky (desktopová verze), na které je viditelná jeho fotografie a osobní informace, které hodlá sdílet.

Informace jsou na desktopové verzi přístupné po najetí kurzoru na obrázek nebo po stisku šipky, která je standardně umístěná pod obrázkem. Nacházíme-li se na cizí profilové stránce, jsou tyto informace viditelné okamžitě. Pod-portál pro desktopy nabízí profilové informace u každé své podstránky v levém sloupci.

V mobilní verzi jsou tyto informace viditelné ihned. Mobilní verze dále na této stránce obsahuje i tlačítka pro rychlý přechod na nástěnku, zprávy nebo přátele. Zároveň je zobrazeno i tlačítko „Zpět“ v levé části horní lišty pro přechod zpět na hlavní menu. I na této stránce jako na všech ostatních se i zde vyskytuje spodní lišta s důležitými oznámeními.

7.3.1 Princip profilů

Profily nebo profilové stránky (s ohledem na druh pod-portálu) a jejich informace jsou stěžejní datovou částí studentského komunikačního portálu. Obsahují veškeré informace o uživateli, které si mohou uživatelé i upravovat.

Na studentském komunikačním portálu rozlišují dva druhy profilů – vlastní profil a cizí profil. Dle toho, na jakém profilu se uživatel momentálně vyskytuje, mu jsou zpřístupněny funkce pro operování s daným profilem.

7.3.2 Detekce profilu

Pro práci s uživatelskými profilem používám proměnnou `$uid`, která udává identifikační číslo uživatele profilu. Tuto proměnnou vyjímám pomocí `$_GET["uid"]` z URI. Shoduje-li se tato proměnná s ID sezení (`$_SESSION["id"]`), tak se uživatel nachází na svém profilu. V opačném případě je uživatel na profilu cizím.

Není-li proměnná `$uid` nastavena (není-li v URI specifikováno `uid`), je uživatel v základu přesunut na svůj profil.

7.3.3 Vlastní versus cizí profil

Na mobilním pod-portálu uživatel, který se nachází na vlastní profilové stránce, vidí svou fotografii a osobní informace. Může přejít na nástěnku, zprávy nebo přátele.

Nachází-li se na cizí profilové stránce, vidí fotografii a informace cizího uživatele. Může přejít na jeho nástěnku, napsat mu soukromou zprávu nebo si ho přidat do přátel (či odebrat).

Na desktopovém pod-portálu uživatel, který se nachází na svém profilu, vidí též svou fotografii a informace. Standardně je mu zpřístupněno ovládací menu, skrze které může přejít na nástěnku, přátele, zprávy a může si upravit profil nebo se odhlásit. Dále na svém profilu vidí i ostatní dříve specifikované prvky.

Na cizím profilu vidí uživatel fotografii a informace cizího uživatele. Ovládací menu mu ovšem nabízí odlišné možnosti – přechod na nástěnku uživatele, poslání soukromé zprávy nebo přidání do přátel (či odebrání). Na cizím profilu dále vidíme 5 posledních návštěv profilu. V základu vidíme i uživatelskou nástěnku, jeho skupiny a jeho pět nejaktivnějších přátel i se skupinami, do kterých patří.

7.3.4 Fotografie uživatele

Každý uživatel má k dispozici po nahrání fotografie na server dvě její verze. První je velká, která se zobrazuje na profilu. Druhá je menší, ta se zobrazuje u zpráv, soukromých zpráv a u komentářů. Jednotlivé fotografie jsem pojmenoval pomocí ID uživatele (tedy dle \$_SESSION["id"]) ve tvaru id.png a id-m.png. Fotografie s afixem -m je označení menší fotografie.

Rozměry fotografií jsem volil tak, aby je bylo možné zobrazit na desktopovém i mobilním zařízení v plných rozměrech.

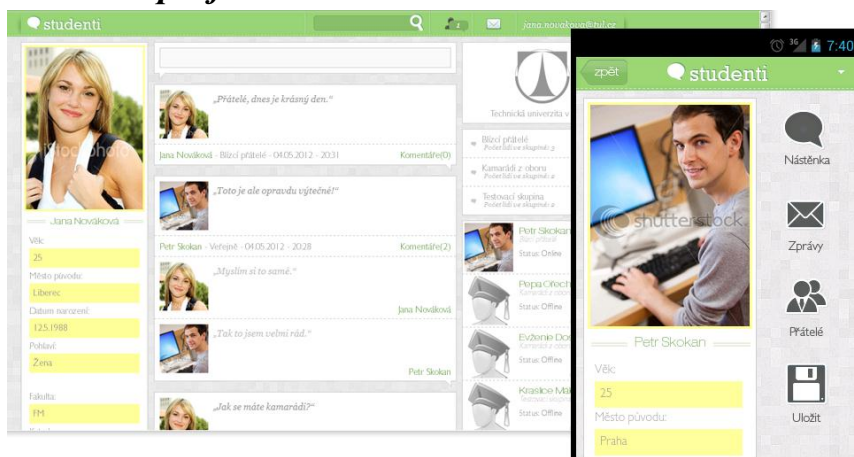
7.3.5 Návštěvy profilu

Zobrazuje-li uživatel cizí profil, ať už na desktopovém nebo mobilním zařízení, je do databáze připsán záznam s jeho návštěvou. Tento fakt je ovšem podmíněn i tím, je-li uživatel veden v posledních pěti návštěvách profilu. Stane-li se tak – nový záznam se nevytvoří. Vše se tedy děje dle následujícího PHP skriptu.

```
if($sid!=$uid){
    $db_call = mysql_query("select count(*) as pocet from (select * from
visits where uid = $uid order by id desc limit 0,5) as t1
where uid = $uid and vid = $sid");
    $db_call_zaz = MySQL_Fetch_Array($db_call);
    $pocet = $db_call_zaz["pocet"];
    if($pocet == 0){
        mysql_query ("insert into visits (uid, vid)
                        values ('".$uid."', '".$sid."');");
    }
}
```

Obrázek 50. Záznam návštěvy profilu

7.3.6 Nastavení profilu



Obrázek 51. Nastavení profilu

Nastavení profilu je přístupné z obou pod-portalů. V obou případech jsou žlutě vyznačeny informace, které lze měnit. Uživatel tedy může měnit jak svou fotografii, tak i všechny informace kromě svého jména. Jména uživatelů jsou používána v modulu hledání a jsou vázána na Shibboleth IdP. Pro jejich informační správnost je potřeba ponechat původní podobu.

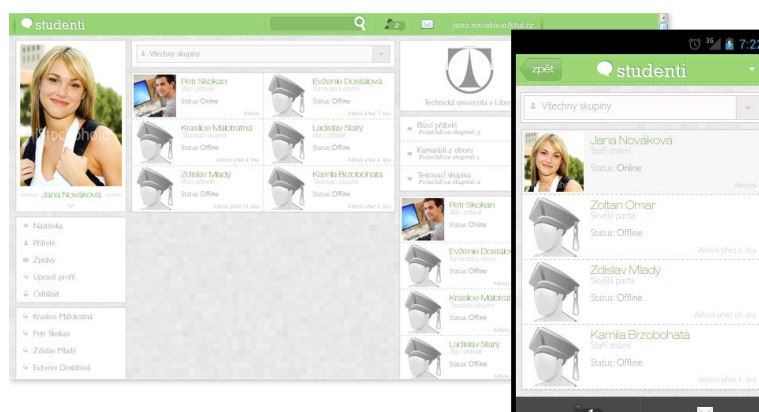
Pro změnu profilové fotografie může uživatel na server nahrát obrázek ve formátu .jpg (.jpeg), .gif a .png. Nahráný obrázek poté pomocí skriptu upravuji na rozměry 180x250 pixelů a ukládám do proměnné. Takto upravený obrázek poté ještě zmenšuji na šířku 75 pixelů a seshora jej ořezávám na výšku 75 pixelů a ukládám do druhé proměnné. Oba tyto obrázky se poté nahrají na server s ID sezení uživatele.

Textové informace měním pomocí nenáročného MySQL dotazu `UPDATE`, který podle ID sezení nalezne požadovaný záznam a upraví jej.

```
mysql_query ("update users set vek = '$vek', mesto = '$mesto',  
             narozeni = '$datum_narozeni', pohlavi = '$pohlavi',  
             fakulta = '$fakulta', katedra = '$katedra',  
             rocnik = '$rocnik', kolej = '$kolej'  
             where id = $sid");
```

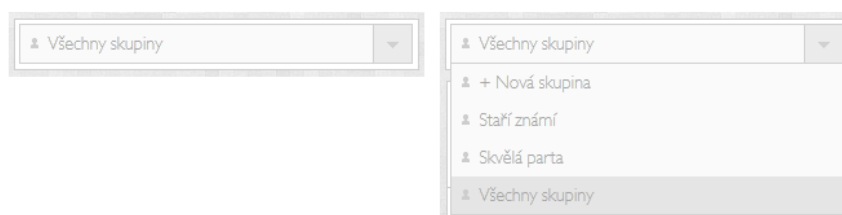
Obrázek 52. MySQL dotaz pro úpravu profilu

7.4 PŘÁTELÉ



Obrázek 53. Přátelé

Tato sekce se ovládá pomocí roletového menu, kde jsou na výběr uživatelské skupiny přátel, které může následně zobrazit. Zobrazit může samozřejmě i všechny skupiny najednou. V každém z těchto případů jsou přátelé seřazeni dle jejich aktivity na portálu.



Obrázek 54. Roletové menu

Z menu dále můžeme vybrat i tvorbu skupiny nové. Formulář, jenž se nám poté vysune, obsahuje pole pro vepsání názvu nové skupiny a tlačítko pro vložení. Během zobrazení nějaké specifické skupiny můžeme tuto skupinu smazat. Touto akcí automaticky zrušíme přátelství se všemi členy této skupiny.

Uživatelské rozhraní na mobilním pod-portálu má jistou odlišnost, kdy se přátelé řadí do jednoho sloupce namísto dvou, jak je tomu u desktopové verze.

7.4.1 Princip přátelství na studentském komunikačním portálu

Uzavírání přátelství je v této webové aplikaci založené, jak již bylo zmíněno, na bázi skupin. Uživatel musí nejprve vytvořit skupinu a poté ji naplnit svými přáteli. Jeden přítel může patřit pouze do jedné skupiny.

Skupiny nemají jen charakter přehlednosti přátel, ale jsou na komunikačním portálu dále využity i při posílání omezených zpráv pro určitou skupinu, kdy je zpráva viditelná pouze těm přátelům, kteří do dané skupiny patří.

7.4.2 Skupiny

Skupiny můžeme vytvářet i mazat. Pro tvorbu skupiny slouží formulář, který se zobrazí po výběru tvorby skupiny z roletového menu. Vzhled formuláře je u mobilních i desktopových verzí identický, jen je odlišně široký.

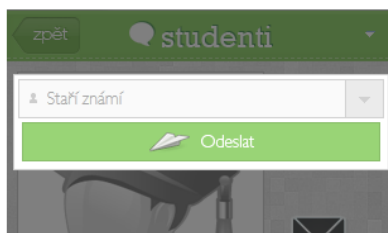
Po odeslání tohoto formuláře, se databázi pošle MySQL dotaz `INSERT`, kterým se do tabulky `groups` vloží název skupiny a id uživatele, který ji vytvořil.

Pro smazání konkrétní skupiny slouží tlačítko vespod stránky. Po smazání skupiny se automaticky zruší přátelství se všemi členy.

7.4.3 Přátelství

Přátelství můžeme zakládat i rušit. Předpoklad pro založení přátelství je existence alespoň jedné skupiny. Bez tohoto předpokladu se na cizím profilu namísto nabídky „Přidat přítele“ zobrazí rada pro založení nové skupiny.

Pro založení přátelství se nejdříve musíme vyskytovat na cizím profilu. Po kliknutí na „Přidat přítele“ stránka ztmavne, ovládací prvky se stanou neaktivní, a zobrazí se roletové menu se skupinami, kdy musíme právě jednu vybrat a uživatel se tak do té skupiny zařadí. Pro demonstraci jsem zvolil jen mobilní verzi, protože formulář pro výběr skupiny je obdobný u obou pod-portálů.

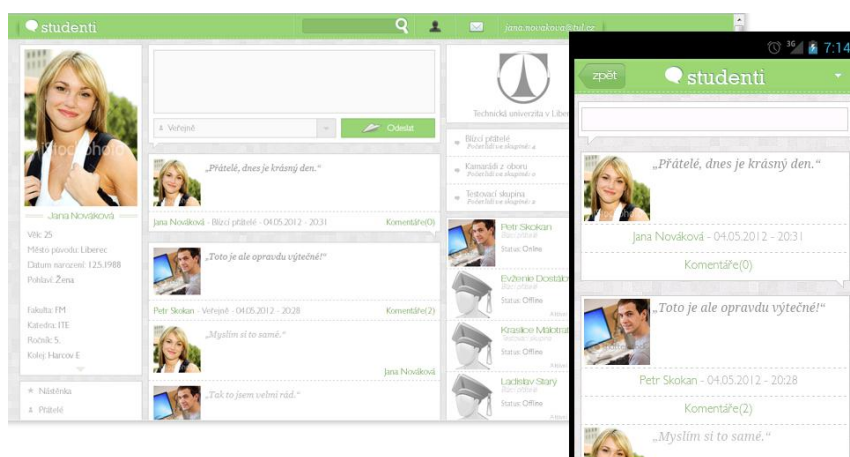


Obrázek 55. Formulář pro přidání přítele – mobilní verze

Odesláním tohoto formuláře se pošle MySQL databázi dotaz `INSERT`, kterým se do tabulky `friends` vloží nový záznam obsahující ID uživatele, ID přítele a ID skupiny. Tabulka `friends` je spojovací tabulkou mezi uživateli, jejich skupinami a přáteli do nich patřícími.

Po přidání přítele se na přítelově profilu namísto „Přidat přítele“ zobrazí nabídka „Odebrat přítele“, pomocí které přítele můžeme odebrat ze svých přátel a ze skupiny, do které patří. V takovém případě se z tabulky `friends` smaže záznam o příteli, který specifikují pomocí všech tří výše zmíněných ID.

7.5 NÁSTĚNKA



Obrázek 56. Nástěnka

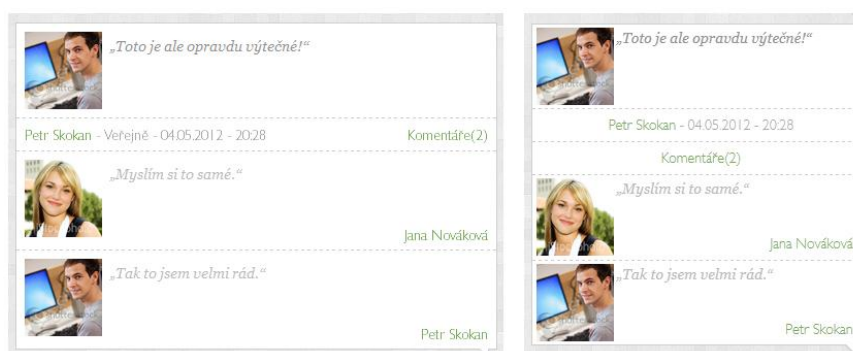
Nástěnka se skládá z formuláře pro psaní zpráv, který musíme nejdříve v obou případech rozkliknout, a ze zpráv. Formulář obsahuje pole pro psaní zpráv, roletové menu pro výběr skupiny a tlačítko pro samotné odeslání.



Obrázek 57. Porovnání formuláře pro vkládání zpráv na obou pod-portálech

Každá zpráva je složena z obrázku uživatele, který ji napsal, jeho jména, samotného těla zprávy, skupiny, do které uživatel patří (nejedná-li se o veřejnou zprávu), z časového údaje, kdy byla zpráva na server vložena a ze dvou posledních komentářů.

V neposlední řadě se skládá i z tlačítka „Komentáře“, které nám ve svém názvu udává i počet komentářů dané zprávy. Po kliknutí na toto tlačítko se rozbalí všechny komentáře zprávy včetně formuláře pro odeslání komentáře nového.



Obrázek 58. Porovnání zpráv na obou pod-portálech

Nástěnka, soukromé zprávy a události mají charakter „nekonečného“ scrollování (projíždění). To znamená, že když najedeme na stránce úplně dolů, tak se nám automaticky načte další obsah.

7.5.1 Princip nástěnky

Nástěnka je část portálu, na které se odehrává většina jeho vnitřní komunikace. Obsahuje všechny veřejné i omezené zprávy (zprávy pro různé skupiny). V zásadě na tomto komunikačním portálu rozlišují dva druhy nástěnek – vlastní nástěnku a cizí. Druh nástěnky detekují identicky, jako tomu bylo při detekci druhů profilů, a to na bázi uživatelského ID `$uid`, které vyjímám z URI (`$_GET["uid"]`).

7.5.2 Vlastní nástěnka

Nachází-li se uživatel na své nástěnce, tak může psát nové zprávy, číst zprávy nebo přidávat komentáře. Forma samotné nástěnky je dána ovšem hlavně tím, jaké zprávy jsou uživateli zobrazeny.

Uživatel vidí všechny zprávy, které napsal on sám a zprávy určené pro některou ze skupin, jejímž je členem. Podmínkou pro zobrazení zprávy určené pro některou ze skupin, jejímž je členem je i to, že musí mít uživatele, který danou skupinu založil, též mezi svými přáteli.

Funkčnost tohoto principu znázorňuje následující MySQL dotaz.

```
select p1.id, p1.uid, p1.gid, p1.cas, p1.zprava from posts as p1
inner join friends as f1 on (p1.uid = f1.uid and
                           (p1.gid = f1.gid or p1.gid = 0) and
                           f1.fid = $sid)
inner join friends as f2 on (f2.uid = $sid and f2.fid = p1.uid)
union select * from posts as p2
where p2.uid = $sid
union select p3.id, p3.uid, p3.gid, p3.cas, p3.zprava from posts as p3
inner join friends as f3 on (f3.uid = $sid and
                           f3.fid = p3.uid and p3.gid= 0)

order by id desc
limit 0,10;
```

Obrázek 59. MySQL dotaz – osobní nástěnka

Tento dotaz se volá pokaždé, když uživatel sjede na WWW stránce nástěnky na úplný konec (s odlišnými parametry `limit` segmentu dotazu). Samotný dotaz je zopakován pomocí AJAX volání a jsou tak načteny další zprávy, které se k obsahu připojí.

7.5.3 Cizí nástěnka

Zprávy na cizí nástěnce může uživatel pouze číst nebo komentovat. Její podoba je opět dána tím, které zprávy jsou uživateli zobrazeny.

V jednoduchém případě, kdy nás nemá uživatel mezi svými přáteli, vidíme pouze jeho veřejné zprávy. Má-li nás ovšem uživatel mezi přáteli, tak vidíme i zprávy, které posílá skupině, do které u něj patříme. V takovém případě nezáleží na tom, zda onoho uživatele máme mezi přáteli i my.

Funkčnost jsem opět řešil skrze MySQL dotaz, který nastíním níže.

```
select p1.id, p1.uid, p1.gid, p1.cas, p1.zprava from posts as p1
inner join friends as f1 on (f1.uid = p1.uid and
                           f1.fid = $sid and
                           f1.uid = $uid and
                           (p1.gid = f1.gid or p1.gid = 0))
union select p2.id, p2.uid, p2.gid, p2.cas, p2.zprava from posts as p2
inner join friends as f2 on (p2.gid = 0)
where p2.uid = $uid
order by id desc
limit 0,10;
```

Obrázek 60. MySQL dotaz – cizí nástěnka

I tento dotaz se volá pomocí technologie AJAX pokaždé, když uživatel sjede na konec WWW stránky.

7.5.4 Vkládání zpráv

Zpráva se na nástěnku vkládá jen za pomoci uživatelova \$sid (\$_SESSION["id"]), abych zamezil možným pokusům o nabourání funkčnosti aplikace. Z databázové tabulky groups vyjímám data o skupinách, které uživatel vytvořil. Pomocí formuláře tak může uživatel danou zprávu adresovat pouze té skupině, které chce. Nechce-li psát zprávu omezeně, nemusí vybírat skupinu a může ji adresovat veřejnosti.

7.5.5 Vkládání komentářů

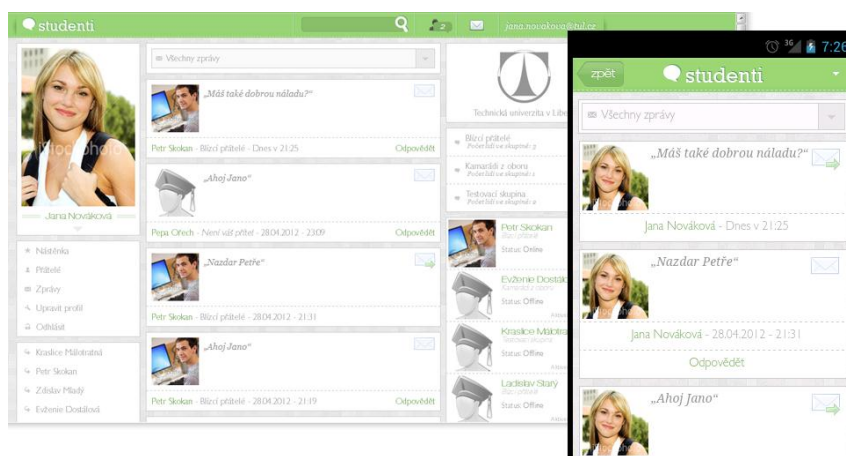
Komentáře můžeme psát bez omezení ke všem zprávám, které máme viditelné. Jejich vkládání je založeno na technologii AJAX, kdy komentář sice vložíme do databáze, a zobrazí se nám ihned mezi ostatními komentáři, ale stránka se celá neobnovuje. Tím pádem uživatel po vložení komentáře zůstane přesně na stejném místě WWW stránky a může bez vyrušení dál číst ostatní zprávy nebo psát jiné komentáře.

Vkládání komentářů demonstruji jen na mobilním pod-portálu, protože princip je stejný v obou verzích.



Obrázek 61. Vkládání komentářů – mobilní verze

7.6 SOUKROMÉ ZPRÁVY



Obrázek 62. Soukromé zprávy

Pomocí roletového menu v horní části můžeme vybírat mezi zprávami, které jsme odeslali my, zprávami, které nám přišly anebo můžeme zobrazit zprávy obou druhů. Vzhled roletového menu je u obou pod-portálů stejný s tím rozdílem, že je pouze jinak široké.

Pod tímto menu vidíme již samotné soukromé zprávy. Jejich struktura je podobná zprávám na nástěnce s tím rozdílem, že neobsahují žádné komentáře. Pokud čteme zprávu od jiného uživatele, můžeme ihned odpovědět pomocí tlačítka „Odpovědět“, kdy se nám rozbalí formulář pro konstrukci nové zprávy.

7.6.1 Princip soukromých zpráv

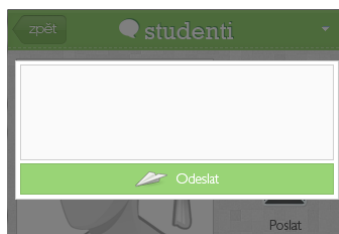
Soukromé zprávy si mezi sebou mohou posílat jak přátelé, tak i uživatelé, kteří mezi sebou přátelství uzavřené nemají. Zprávy lze posílat jak po zobrazení profilu cizího uživatele, tak i z prostředí určeného pro zprávy. Všechny soukromé zprávy jsou uloženy v tabulce `pms`. Dostane-li uživatel soukromou zprávu od uživatele, kterého nemá v přátelích, je u něj uveden popisek, že se jedná o cizího člověka.

Při zobrazení uživatelského rozhraní soukromých zpráv se u uživatele v databázové tabulce `users` automaticky aktualizuje na momentální čas Unixová časová stopa, která slouží pro upozornění na nové zprávy.

7.6.2 Posílání soukromých zpráv z cizího profilu

Zprávy můžeme posílat z cizího profilu po kliknutí na tlačítko v ovládacím menu „Poslat zprávu“ (desktopová verze) nebo po kliknutí na obrázek obálky s popiskem „Poslat“ (mobilní verze). V obou případech pak stránka ztmavne, ovládací prvky se stanou neaktivní a zobrazí se formulář pro odeslání soukromé zprávy. Po kliknutí na tlačítko odeslat formulář zmizí a do tabulky `pms` se pomocí MySQL dotazu `INSERT` vloží nový záznam s ID uživatele a příjemce, časem odeslání a obsahem zprávy.

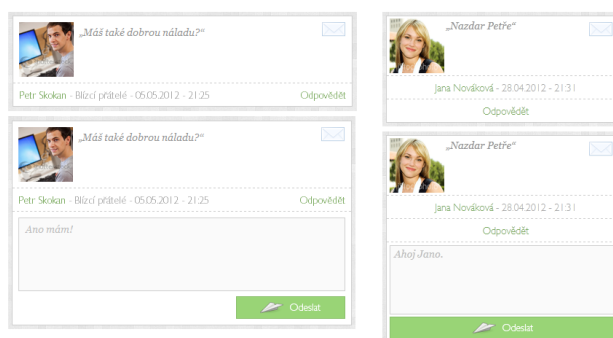
Formulář pro odeslání zprávy má stejný charakter v obou verzích pod-portálu, proto demonstruji jen ten, který se vyskytuje v mobilní verzi.



Obrázek 63. Formulář pro odeslání soukromé zprávy – mobilní verze

7.6.3 Odpověď na soukromou zprávu

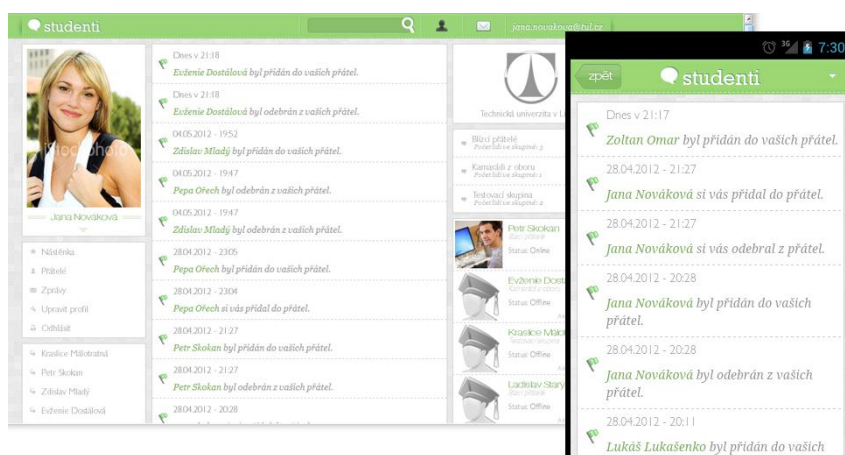
Na soukromé zprávy můžeme odpovídat buď přímo z profilu cizího uživatele odesláním nové zprávy, nebo můžeme využít tlačítka pro rychlou odpověď „Odpovědět“, které se vyskytuje u všech cizích zpráv v uživatelském rozhraní pro soukromé zprávy. Po kliknutí na tlačítko „Odpovědět“ se vysune formulář pro konstrukci nové soukromé zprávy.



Obrázek 64. Rozdíl formulářů pro odpověď na soukromou zprávu v obou pod-portálech

Jako tomu bylo v předchozím případě při odesílání soukromých zpráv, po kliknutí na tlačítko odeslat formulář zmizí a do tabulky `pms` se pomocí MySQL dotazu `INSERT` vloží nový záznam s ID uživatele a příjemce, časem odeslání a obsahem zprávy.

7.7 UDÁLOSTI



Obrázek 65. Události

Uživatelské rozhraní událostí obsahuje seznam důležitých činností na portálu. Každý záznam se skládá z času vytvoření, zainteresované osoby a z údaje, který je s ní spojován. Seznam je na obou pod-portálech zobrazen identicky, pouze má jinou šířku.

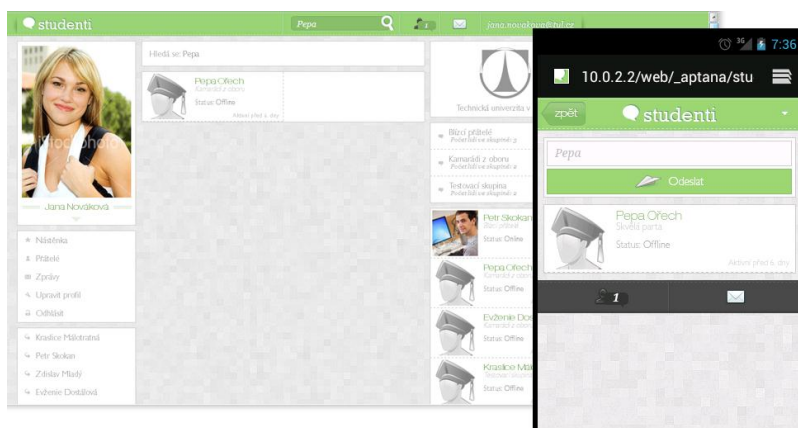
7.7.1 Princip událostí

Události jsou zprávy generované systémem při nějaké významné činnosti uživatele nebo jeho přátel. Jmenovitě je to např. událost, kdy si uživatele někdo jiný přidá do přátel. Události jsou řazeny dle času, kdy k nim došlo.

Při zobrazení uživatelského rozhraní událostí se u uživatele v databázové tabulce `users` automaticky aktualizuje na momentální čas Unixová časová stopa, která slouží pro upozornění na nové události.

Stejně jako nástěnka a soukromé zprávy jsou i události založeny na bázi „nekonečného“ scrollování (projíždění). Když najedeme na stránce úplně dolů, tak se nám automaticky načte další obsah, tedy další záznamy v seznamu událostí.

7.8 HLEDÁNÍ



Obrázek 66. Hledání

Modul hledání se na desktopovém pod-portálu nachází na horní liště vlevo od upozornění. Vepsáním údaje a stisknutím klávesy „Enter“ jej aktivujeme. Poté jsme přesunuti na stránku s vyhodnocenými výsledky.

V mobilní verzi je tento formulář přístupný skrze hlavní menu. Po jeho vyplnění a odeslání zůstává uživatel na stejné stránce, kde se vyhodnotí a zobrazí výsledky.

Výsledky tohoto procesu jsou nalezení uživatelé, kteří jsou seřazeni dle jejich aktivity na portálu. Rozdíl v mobilní verzi je takový, že se výsledky řadí do jednoho sloupce namísto dvou, jako je tomu u verze desktopové.

7.8.1 Princip modulu hledání

Modul hledání, který jsem vytvořil, hledá pouze studenty, kteří tento webový portál alespoň jednou využili. Zamezíme tím hledání studentů, kteří portál nevyužívají.

Hledaný výraz se hledá jako podřetězec v uživatelských jménech a nerozlišují se malá/velká písmena. Samotné hledání je vytvořeno pomocí následujícího MySQL dotazu.

```
select * from users
where UPPER(jmeno) like '%"strtoupper($hledat)."' and id != $sid
order by aktivni desc;
```

Obrázek 67. Modul hledání

Hledání je zároveň omezeno tak, aby jeden z výsledků nebyl právě sám uživatel.

7.9 NEPŘÍMO SOUVISEJÍCÍ SOUČÁSTI

Jelikož jsem vyvinul multi-platformní portál, musel jsem se zaobírat i částmi, které se samotnou funkčností portálu sice nemají co dočinění, ale zpříjemňují užívání samotného portálu.

7.9.1 Touch ikona

Při prohlížení webové stránky je uživateli mimo klasické `favicon` ikony, která je v prohlížeči zobrazena vedle URI webové stránky, předložena též tzv. „touch“ ikona, jejíž funkčnost se projeví právě až u mobilních zařízení. Když si uživatel přidá odkaz na studentský portál na plochu svého mobilního zařízení, tak je tento odkaz vizualizován právě pomocí této ikony. Takové ikoně se zpravidla, jak to u mobilních zařízení bývá zvykem, při jejím vyobrazení i zakulatí rohy.

7.9.2 Fonty

Při vývoji studentského komunikačního portálu jsem z estetických důvodů používal i fonty, které se v základu nevyskytují na žádné platformě. Jedná se o Gill Sans MT Pro - Light a Helvetica Neue LT Pro - ThEx.

Pro správné zobrazení zmíněných písem na všech zařízeních jsem zahrnul těchto pět typů fontů: .eot, .otf, .svg, .ttf, .woff

Mimo těchto fontů jsem používal i font Georgia, ten se vyskytuje na většině moderních zařízení. Kdyby ovšem uživatel tento font na svém zařízení neměl, tak jsem pomocí CSS ošetřil jeho vizualizaci i dalšími zástupnými fonty.

7.10 ODHLÁŠENÍ

Uživatel se může z komunikačního portálu odhlásit prakticky ve kterýkoli okamžik. Odhlásit se může na obou pod-portálech dvěma způsoby – pomocí vysouvacího menu na horní liště nebo pomocí tlačítka odhlášení, které je přístupné z ovládacího menu (desktopová verze) nebo z hlavního menu (mobilní verze).

7.10.1 Princip odhlášení

Odhlášení probíhá zrušením uživatelského ID sezení (\$_SESSION[„id“]) a přesměrováním na úvodní přihlašovací stránku. Tuto činnost nejlépe znázorní následující PHP skript.

```
<?php
    session_start();
    session_unset();
    session_destroy();
    session_write_close();
    setcookie(session_name(), '', 0, '/');
    session_regenerate_id(true);
    header('Location: index.php');
?>
```

Obrázek 68. Odhlášení – PHP

Tento skript se volá při odhlášení jak z mobilního pod-portálu, tak i z desktopového.

ZÁVĚR

Vzhledem k tomu, že v dnešní době lidé používají mobilní zařízení již od raného dětství, lze předpokládat, že důležitost WWW stránek přístupných i skrze tato mobilní zařízení bude nabývat na čím dál větším významu. Vývoj takových stránek, které jsou přístupné skrze širokou škálu zařízení, je ovšem problematikou dosti náročného charakteru. Smutným faktem tak zůstává, že v současnosti neexistuje žádný souhrn pravidel nebo tipů pro tvorbu takových aplikací. V této diplomové práci jsem se tento zásadní nedostatek snažil napravit. S jejím obsahem by si každý webový designer měl vytvořit základní přehled o tom, jak takovou webovou aplikaci, která bude uzpůsobená i pro mobilní zařízení, vytvořit.

Nejprve jsem specifikoval důležité moderní technologie, jichž lze pro vývoj WWW aplikací v současnosti využít. Vybíral jsem zároveň takové technologie, které mají co dočinění s touto diplomovou prací, a které se mi zdají být nadčasové. Většina z nich dle mého úsudku zažije svou největší slávu až za pár měsíců či let, kdy problematika tvorby WWW aplikací pro široké spektrum zařízení začne gradovat na její důležitosti.

Teoretickou část práce jsem započal analýzou, kdy jsem zprvu objasnil stav, ve kterém se nachází současné webové stránky, jejich chyby a nedostatky. Jako cíl analýzy jsem si stanovil souhrn vlastností zařízení, které nějakým způsobem ovlivňují vizualizaci WWW stránek. Samotnou analýzu jsem proto provedl ve třech krocích, kdy jsem nejprve analyzoval webové aplikace vzhledem k různým platformám z hlediska přístupnosti, poté z pohledu použitelnosti a v neposlední řadě i rychlosti. Výstup této analýzy mi v diplomové práci sloužil jako zásadní nástroj pro další rozbor.

Další etapou byla specifikace přístupů k multiplatformním webovým aplikacím. V této pasáži jsem se zaměřil na dva hlavní faktory, které ovlivňují vizualizaci webových stránek nejvíce, a které nelze bez patřičného rozboru náležitě správně ošetřit. Na mysli mám rozlišení a DPI zařízení. Pracoval jsem s těmito činiteli v rámci různých možností jejich uzpůsobování pro vizualizaci WWW stránek. Po srovnání všech experimentálních přístupů jsem zjistil, že ač existuje řešení aplikovatelné pro všechny druhy displejů, jeho implementace je značně komplikovaná a víceméně i kontraproduktivní.

Po analýze a specifikaci přístupů jsem se mohl již zaměřit na samotné zhodnocení vlastností zařízení. Tento segment diplomové práce jsem strukturoval na bázi výstupu z analýzy. Samotné zhodnocení vlastností jsem pojal i jako ujasnění jejich optimalizace vzhledem k možným problémům, které tyto faktory mohou způsobovat. Pokryl jsem všechny vlastnosti zařízení, které nějakým způsobem ovlivňují vizualizaci WWW aplikací a to z hlediska hardwaru, softwaru i dat.

Následně jsem započal druhou zásadní partii diplomové práce – praktickou část. Její první oddíl jsem věnoval návrhu webové aplikace, při kterém jsem upotřebil znalosti nabyté z části teoretické. Nejprve jsem pro lepší pochopení situace objasnil

obecnou architekturu celého systému, který jsem implementoval. Podrobnější podobu aplikace ovšem znázorňuje až další návrh, který se zabývá strukturou studentského komunikačního portálu jako takového. V poslední návrhové části jsem se pak zaměřil i na databázi a její strukturu.

Další oddíl praktické části (a zároveň poslední oddíl diplomové práce) jsem věnoval jednotlivým částem portálu. Popsal jsem uživatelské rozhraní všech podstránek a rozdílů přístupu z různých zařízení. U nejdůležitějších částí jsem pak demonstroval i jejich funkční činnost z hlediska programátora.

Studentský komunikační portál jsem dal k testování vybrané skupině jedenácti studentů z Technické univerzity v Liberci, kteří jej odzkoušeli z mobilních i desktopových zařízení. Mobilní verze portálu byla studenty hodnocena kladně, a to zejména díky zpřístupnění stejných funkcí, jež obsahuje verze desktopová. Ohlasy na uživatelské rozhraní, které je různé na obou platformách, a funkce portálu byly vždy pozitivní. Jediné výhrady byly k barevnému provedení designu, který jsem pak dle názorů upravoval. Portál by se tak v budoucnu mohl například kromě dalších funkčních činností rozšířit i o výběr barevných témat.

Cíle mého zadání diplomové práce byly splněny, přičemž jsem v konečném důsledku jejich řešení zašel ještě dál a z práce vyplynuly výsledky, které lze použít nejen pro desktopová a mobilní zařízení současnosti, ale např. i pro projektory, televize, archivní stroje nebo zařízení budoucnosti, o kterých si netroufám ani bájít. Dodržením specifikovaných postupů a tipů ujasněných v této práci je zajištěna správná vizualizace webových stránek na všech takových zařízeních.

SEZNAM POUŽITÉ LITERATURY

- [1] LAWSON, Bruce a Remy SHARP. *Introducing HTML5*. Vydání první. Berkeley, CA: New Riders, 2011. ISBN 03-216-8729-9.
- [2] LUBBERS, Peter, Brian ALBERS a Frank SALIM. *Pro HTML5 programming: powerful APIs for richer Internet application development*. Vydání první. New York: Apress, 2010. ISBN 14-302-2790-7.
- [3] GASSTON, Peter. *The book of CSS3: a developer's guide to the future of web design*. Vydání první. San Francisco: No Starch Press, 2011. ISBN 15-932-7286-3.
- [4] MARCOTTE, Ethan. *Responsive web design* [online]. Vydání první. New York: A Book Apart, 2011 [cit. 2012-05-01]. ISBN 978-098-4442-577. Dostupné z: <http://www.abookapart.com/products/responsive-web-design>.
- [5] RIVOAL, Florian et al. Media Queries. *W3.org* [online]. 2012 [cit. 2012-04-30]. Dostupné z: www.w3.org/TR/css3-mediaqueries.
- [6] FLANAGAN, David. *JavaScript: the definitive guide*. Vydání šesté. Sebastopol, CA: O'Reilly, 2011. ISBN 05-968-0552-7.
- [7] RESIG, John. *JQuery: the write less, do more, JavaScript library* [online]. 2012 [cit. 2012-05-02]. Dostupné z: www.jquery.com.
- [8] DARIE, Cristian, Bogdan BRINZAREA a Filip CHERECHES-TOSA. *AJAX and PHP building responsive web applications*. Vydání první. Birmingham, U.K.: Packt Pub, 2006. ISBN 19-048-1182-5.
- [9] SHIBBOLETH. *Shibboleth* [online]. 2012 [cit. 2012-05-01]. Dostupné z: <http://shibboleth.internet2.edu>.
- [10] SATRAPA, Pavel. Shibboleth – identifikujte se jen jednou. *Lupa.cz* [online]. 2005 [cit. 2012-05-01]. Dostupné z: <http://www.lupa.cz/clanky/shibboleth>.
- [11] NETCRAFT. *May 2012 Web Server Survey* [online]. 2012 [cit. 2012-05-01]. Dostupné z: <http://news.netcraft.com/archives/2012>.
- [12] MICROSOFT CORPORATION. *PHP on IIS7: the official Microsoft IIS site* [online]. 2012 [cit. 2012-05-01]. Dostupné z: <http://php.iis.net>.
- [13] BURKS, Michael, Patrick LAUKE, Jim THATCHER, Richard RUTTER a Cynthia WADDELL. *Web accessibility: web standards and regulatory compliance*. Vydání první. New York: Springer-Verlag, 2006. ISBN 978-159-0596-388.
- [14] COLBORNE, Giles. *Simple and usable: Web, mobile, and interaction design*. Vydání první. Berkeley, CA: New Riders, 2011. ISBN 03-217-0354-5.
- [15] KING, Andrew B. *Speed up your site: Web site optimization*. Vydání první. Indianapolis, Ind.: New Riders, 2003. ISBN 07-357-1324-3.
- [16] BROGAN, Michael. Install Shibboleth Service Provider on Windows and IIS – IAM – UW Information Technology Wiki. *Wiki.cac.washington.edu* [online]. 2012 [cit. 2012-05-05]. Dostupné z: <https://wiki.cac.washington.edu/display/infra/Install+Shibboleth+Service+Provider+on+Windows+and+IIS>.

PŘÍLOHY NA CD

- ❖ Abstrakt ve formátu DOCX a PDF
- ❖ Vlastní diplomová práce ve formátu DOCX a PDF
- ❖ WWW aplikace vytvořená v praktické části této diplomové práce
- ❖ Grafické soubory WWW aplikace ve formátu PSD
- ❖ MySQL databáze